Underwater Vehicles

Duc Cuong Vu

Motion Control and Applied Robotics Laboratory,
School of Electrical and Electronic Engineering,
Hanoi University of Science and Technology

Version: 1.0

Last updated: January 9, 2025

Table of Contents

Table of Contents

UUV - Unmanned Underwater vehicle

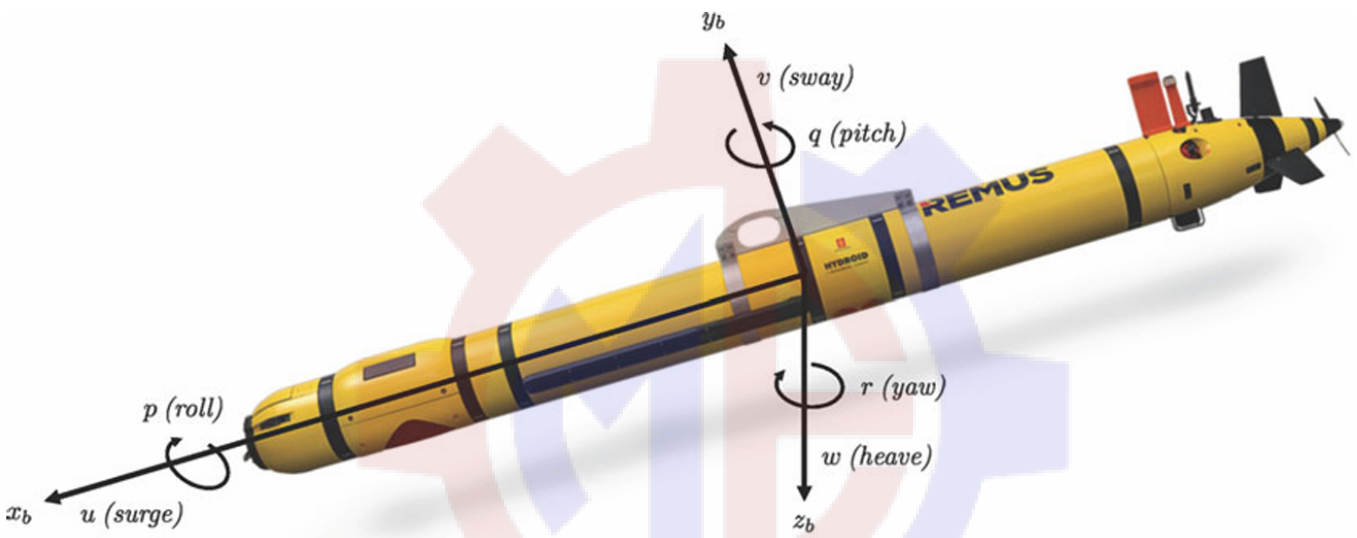
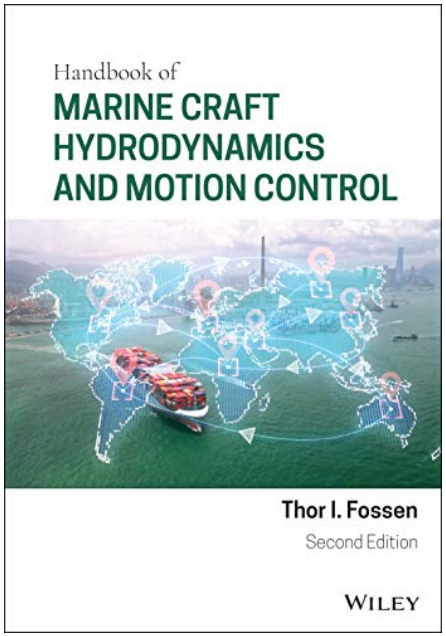
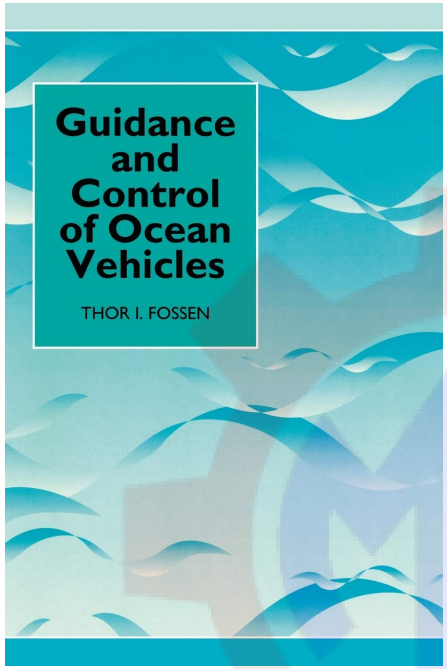


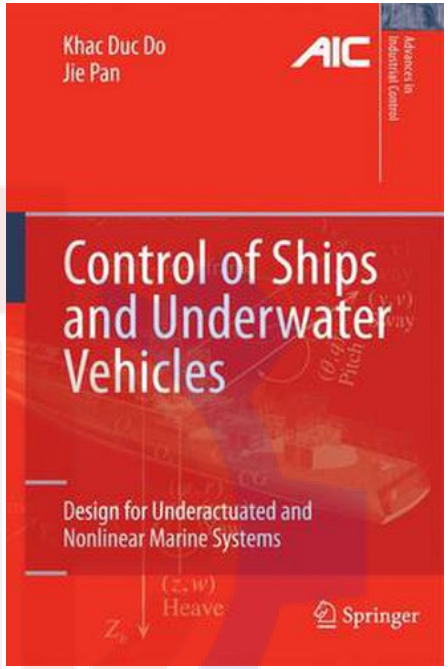
Figure: 6-DOF motion variables for a UUV/AUV

Handbooks & References:

- Thor I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, Wiley, 2011.
- Thor I. Fossen, *Guidance and Control of Ocean Vehicles*, Wiley, 1994.







Ocean Engineering 218 (2020) 108179

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Finite-time extended state observer based nonsingular fast terminal sliding mode control of autonomous underwater vehicles

Nihad Ali, Isaac Tawiah, Weidong Zhang *

Department of Automation, Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Keywords:
Autonomous underwater vehicles
Nonsingular fast terminal sliding mode control
Finite-time observer
Trajectory tracking

ABSTRACT

In this paper, a finite-time extended state observer-based nonsingular fast terminal sliding mode control (NFTSMC) is proposed for the trajectory tracking of autonomous underwater vehicles (AUVs) with various hydrodynamic uncertainties and external disturbances. First, a proportional-integral velocity variable based third-order fast finite-time extended state observer is designed to estimate the lumped uncertainties and their first derivatives. The concepts of finite-time stability and a new proposition are utilized to prove the finite-time uniformly ultimately boundness of the proposed observer. Then, based on the disturbance estimation, an NFTSMC is developed for AUVs. The inverse tangent function and linear control term are used to provide robustness against estimation error and avoid chattering in the thrusters. The position and attitude trajectories are proved to be fast finite-time stable via the Lyapunov theorem. To verify the theoretical analysis, comparative numerical simulations are provided to demonstrate the effectiveness of the proposed control scheme.

1. Introduction

Autonomous underwater vehicles (AUVs) are widely used to accomplish numerous undersea operations such as monitoring and surveillance, petroleum exploration, underwater rescue, marine meteorological observation, deep-sea archaeological research, and oceanographic research (Paull et al., 2014; Jacobi, 2015; Yan et al., 2019a; Kumar and Rani, 2020). To perform the above operations and acquire quality data, the accurate trajectory tracking control of AUVs is required. Since the AUVs are always subject to highly nonlinear and strongly coupled dynamic model which operates in an uncertain oceanic environment and possess model uncertainties, the conventional linear controllers may be unsuitable to cope with these problems. The existence of such uncertainties and various environmental disturbances brings adverse effects on the attitude and position tracking performance, rendering great challenges in the control design.

Over the last decade, various advanced approaches like sliding mode control (SMC) (Yan and Yu, 2018), backstepping (Cho et al., 2020), adaptive control (Hassanein et al., 2016), model predictive control (Khan et al., 2020), and disturbance observer-based feedback control (Mukherjee et al., 2015) have been proposed by the research community to achieve precise position control performance. Based on the convergence analysis, these methods can be classified as exponential or asymptotic stable. Among these control methodologies, SMC has received considerable attention because of its remarkable features of strong robustness, order reduction, and design simplification (Utkin, 1992). Owing to these features, SMC was designed for underwater vehicles to compensate the external disturbances and hydrodynamic uncertainties (Yoerger and Slotine, 1991; Huang and Yang, 2019; Elmokadem et al., 2016; Yan et al., 2019b). The first experimental validation of adaptive SMC was performed by Yoerger and Slotine (1991) for trajectory tracking control of AUV and discuss their implementation issues. In practical implementations, SMC suffers from unwanted chattering phenomenon due to a discontinuous feedback term in the control signal. This switching action occurs at high frequency and may excite the unmodeled dynamics leading to frequent oscillations in the speed of the thrusters, which result in large energy consumption and even may damage the thrusters (Huang and Yang, 2019). The chattering problem can be reduced by approximating the discontinuous terms by continuous functions like saturation function, arctan function, and hyperbolic tangent function (Huang and Yang, 2019; Elmokadem et al., 2016; Labbadi and Cherkaoui, 2019a). The traditional SMC methods adopt the conventional linear switching surfaces and ensure the asymptotic convergence of attitude and position errors to equilibrium points.

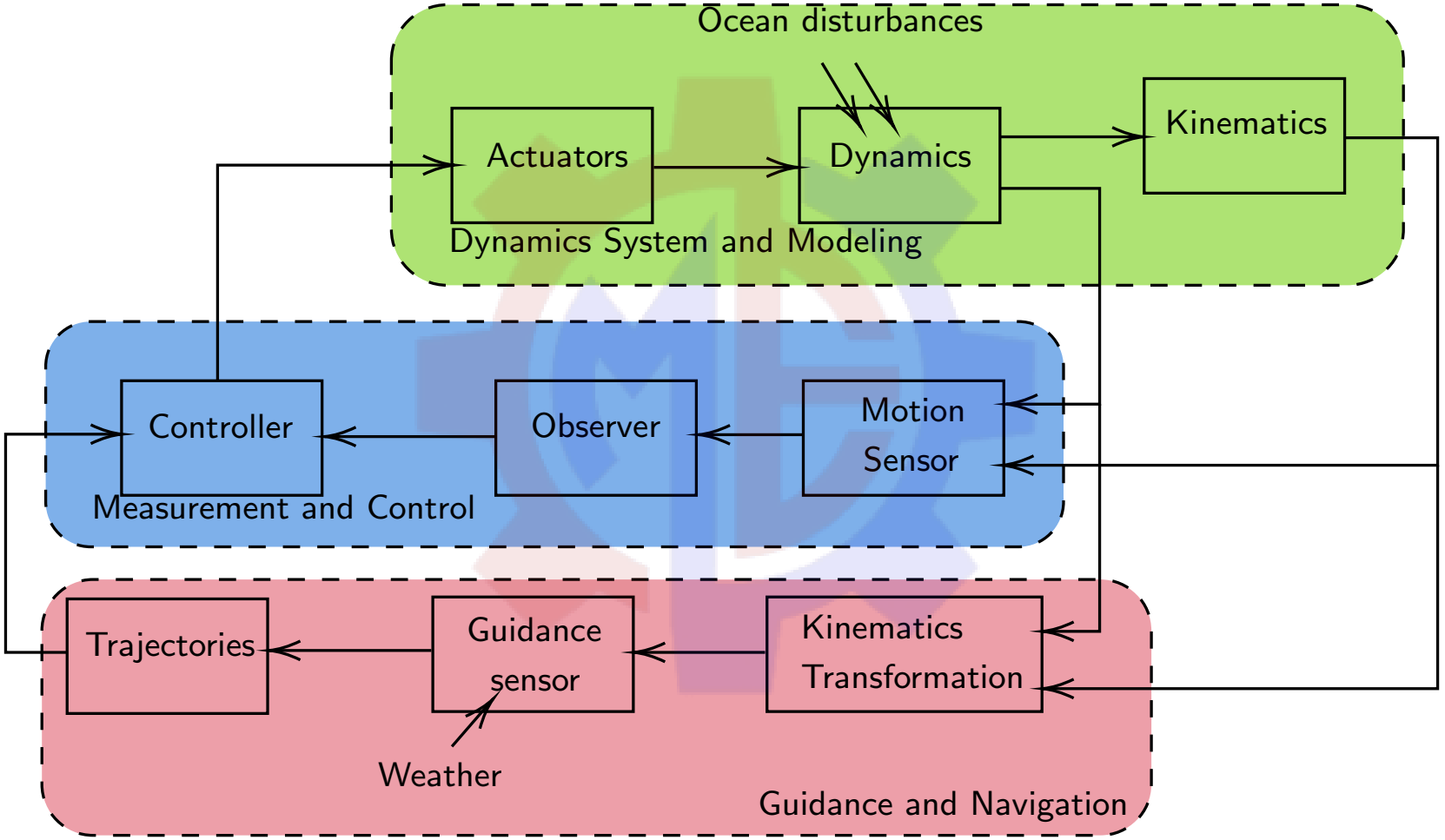
Finite-time control is a particular case of the asymptotic tracking control, and it can force the system trajectories to reach their equilibrium point in a finite time (Hu et al., 2019). The fractional power terms in the finite-time control law can accelerate the convergence

* Corresponding author.
E-mail address: nihad@sjtu.edu.cn (N. Ali), isaac_k_tawiah@sjtu.edu.cn (I. Tawiah), wdzhang@sjtu.edu.cn (W. Zhang).

https://doi.org/10.1016/j.oceaneng.2020.108179
Received 18 May 2020; Received in revised form 20 August 2020; Accepted 27 September 2020
Available online 6 October 2020
0029-8018/© 2020 Elsevier Ltd. All rights reserved.

Handbooks & References:

- Review articles, Research articles, Handbook, ...
- IEEE Explore, Springer, Elserview, Wiley, Taylor & Francis, ...


Ocean disturbances
Actuators
Dynamics
Kinematics
Dynamics System and Modeling
Controller
Observer
Motion
Sensor
Measurement and Control
Trajectories
Guidance
sensor
Kinematics
Transformation
Weather
Guidance and Navigation

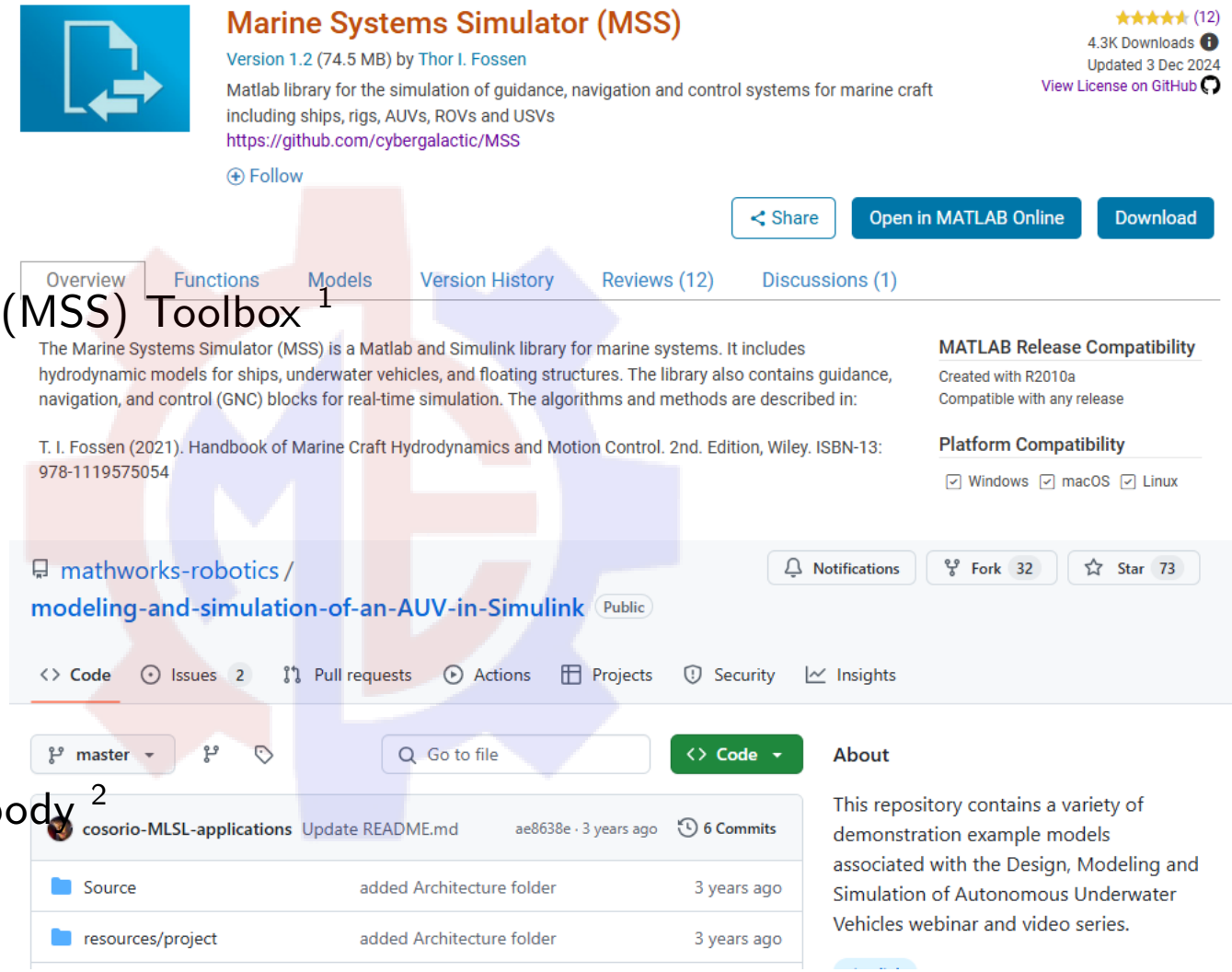
Degree of Freedom classifications under purpose

- 1 DOF $\rightarrow$ model can be used to design the forward speed controller and heading autopilots and damping system
- 3 DOFs $\rightarrow$ describe the horizontal plane, longitudinal, and lateral models.
- 4 DOFs $\rightarrow$ describe the motion in the horizontal plane with heading autopilots and system damping
- 6 DOFs $\rightarrow$ describe the fully coupled equation of motion used for simulation and prediction.

Naval Architecture

- Maneuvering theory: moving at positive constant speed in calm water
- Seeking theory: motion of ships at zero or constraint speed in waves, which can be analyzed. The hydrodynamic coefficient and wave forces are computed as a function

Marine Systems Simulator (MSS) Toolbox [1]

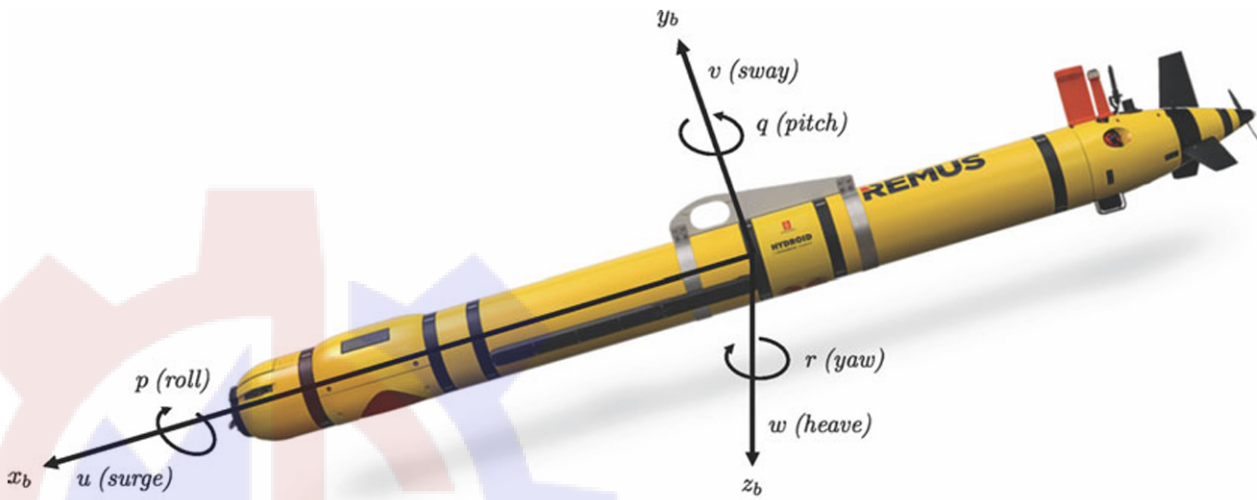


MATLAB Simscape Multibody [2]

[1] T. I. Fossen and T. Perez (2004). Marine Systems Simulator (MSS). URL: https://github.com/cybergalactic/MSS

[2] https://github.com/mathworks-robotics/modeling-and-simulation-of-an-AUV-in-Simulink

Introduction



Overview of this seminar

- Fundamental knowledge
- 6-DOF model of a Rigid Body
- Ocean dynamics
- Environmental disturbances
- An example of the ODIN
- Guidance - Navigation - Control
- Numerical simulations
- Quasi-Physical Simulation for UV motion control
- An excample for GNC-integrated Quasi-Physical simulation

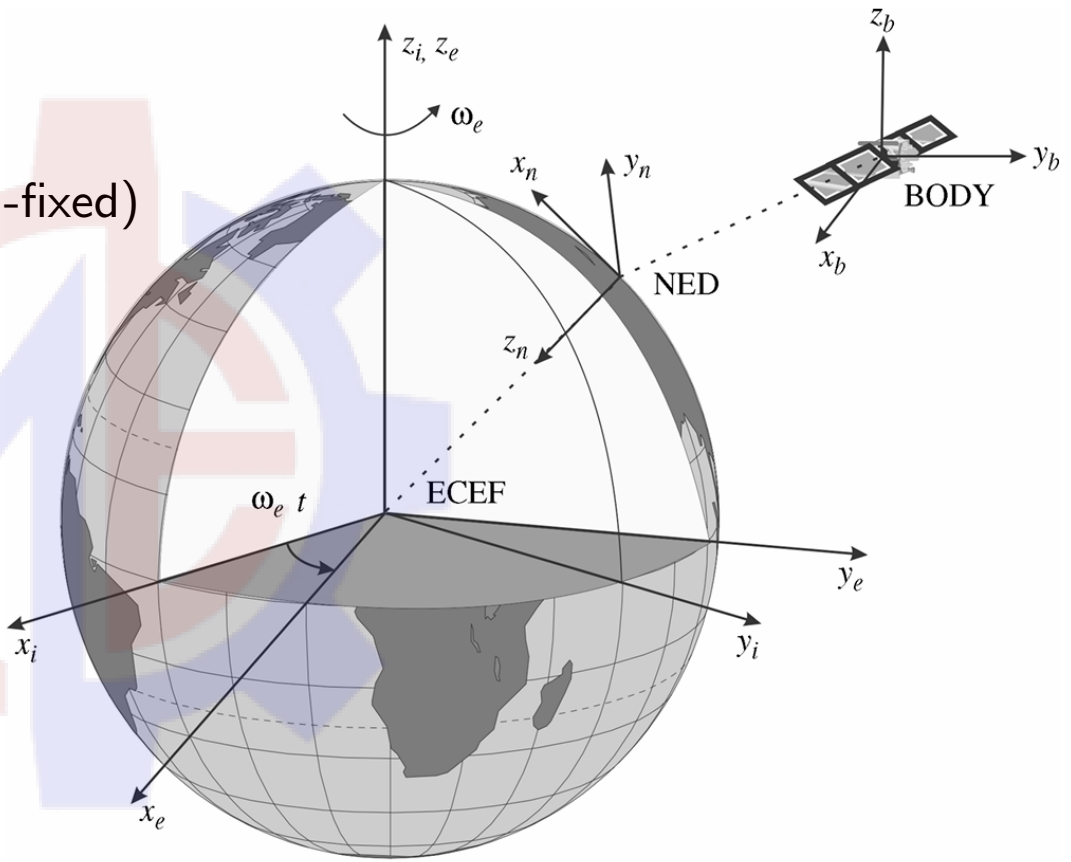
Table of Contents

Earth-Centered reference frames

- ECI - $\{i\}$ - (Earth-centered inertial)
- ECEF - $\{e\}$ - (Earth-centered Earth-fixed)

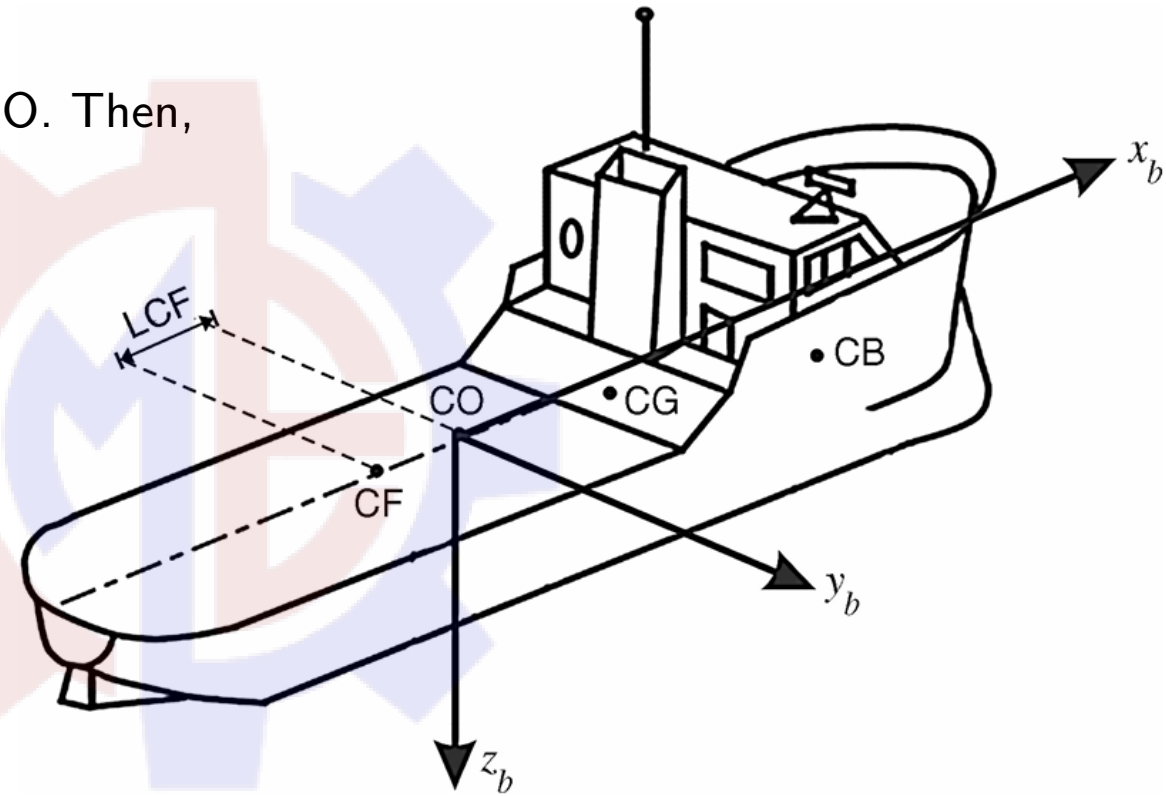
Geographic reference frames

- NED - $\{n\}$ - (North-East-Down)
- BODY - $\{b\}$ - (Craft-fixed)

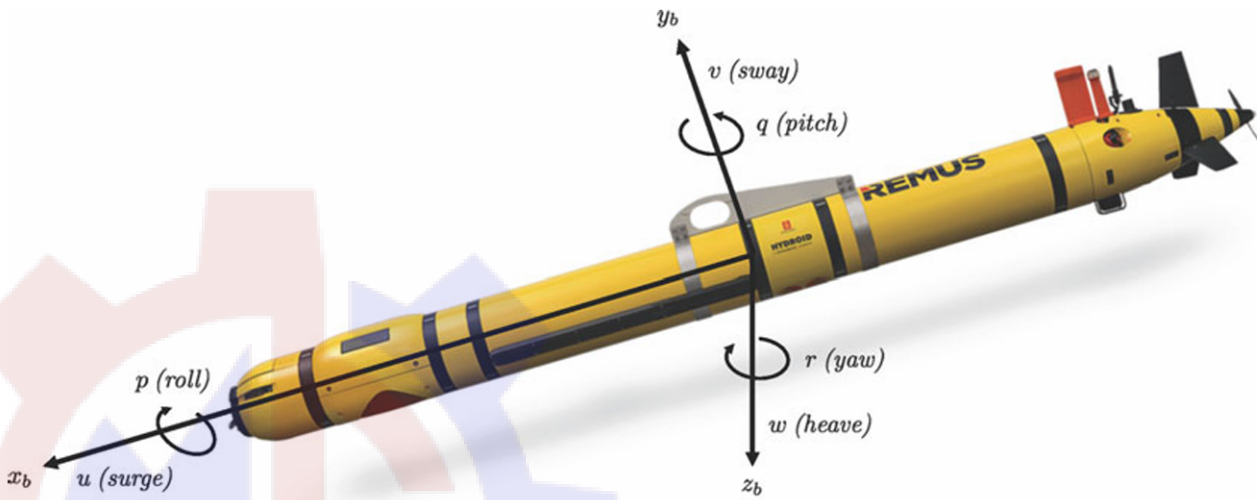


The origin of the BODY frame is CO. Then,

- CG - Center of Gravity
- CB - Center of Buoyancy
- CF - Center of flotation

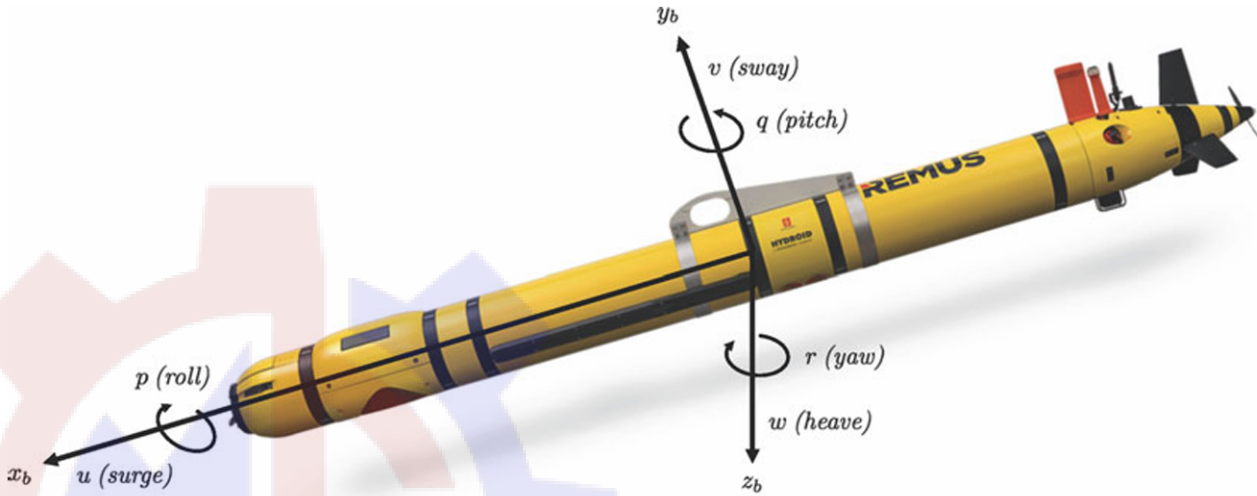


The center of flotation is the centroid of water plane area A_{wp} in calm water.



Notation of SNAME (1950)

- Motion of x-direction (surge) – x, controlled by X [N]
- Motion of y-direction (sway) – y, controlled by Y [N]
- Motion of z-direction (heave) – z, controlled by Z [N]
- Rotation of x-direction (roll) – ϕ, controlled by K [Nm]
- Rotation of y-direction (pitch) – θ, controlled by M [Nm]
- Rotation of z-direction (yaw) – ψ, controlled by N [Nm]



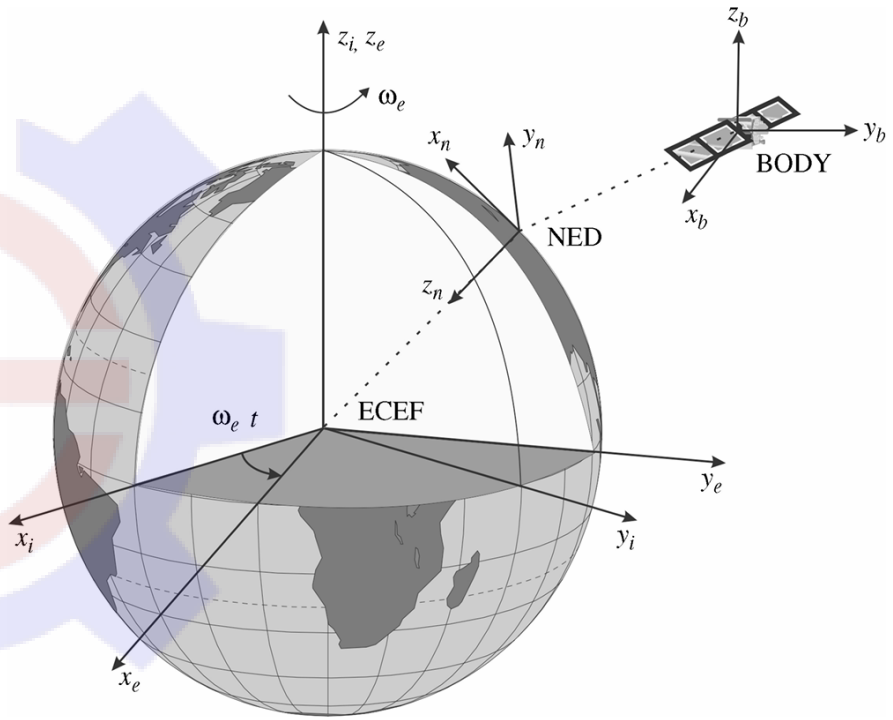
General coordinate

- $\eta = \begin{bmatrix} (p^n_{b/n})^\top & \Theta^\top_{nb} \end{bmatrix}^\top$, and $p^n_{b/n} = \begin{bmatrix} N & E & D \end{bmatrix}^\top$, $\Theta_{nb} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\top$. In addition, the relative position of craft could be represented in frame ECI.
- $\nu = \begin{bmatrix} (v^b_{b/n})^\top & (\omega^b_{b/n})^\top \end{bmatrix}^\top$, and $v^b_{b/n} = \begin{bmatrix} u & v & w \end{bmatrix}^\top$, $\omega^b_{b/n} = \begin{bmatrix} p & q & r \end{bmatrix}^\top$
- $\tau = \begin{bmatrix} \tau_1^\top & \tau_2^\top \end{bmatrix}^\top$, and $\tau_1 = \begin{bmatrix} X & Y & Z \end{bmatrix}^\top$, $\tau_2 = \begin{bmatrix} K & M & N \end{bmatrix}^\top$

Notice that η denotes the position and orientation vector with coordinates in the earth-fix frame. ν denotes the linear and angular velocity vector with coordinates in the body-fixed frame and τ denotes the force and torque acting on the vehicle.

Frames transformation

- Euler angles
- Rotation matrices
- Skew-Symmetric of a matrix
- Cross product
- Euler's theorem on rotation
- Linear/Angular velocity transformation
- Rotation matrix differential
- ...



In another perspective, **quaternion** representation is also a good option!

The 6 DOFs and 3 DOFs kinematic equations can be expressed in the following

6 DOFs kinematic equations

$$\dot{\eta} = J_\Theta(\eta)\nu \tag{1}$$

that is equivalent to

$$\begin{bmatrix} \dot{p}^n_{b/n} \\ \dot{\Theta}_{nb} \end{bmatrix} = \begin{bmatrix} R^n_b & 0 \\ 0 & T_\Theta(\Theta_{nb}) \end{bmatrix} \begin{bmatrix} v^b_{b/n} \\ \omega^b_{b/n} \end{bmatrix} \tag{2}$$

where R^n_b and $T_\Theta(\Theta_{nb})$ denote the rotation and transformation matrices.

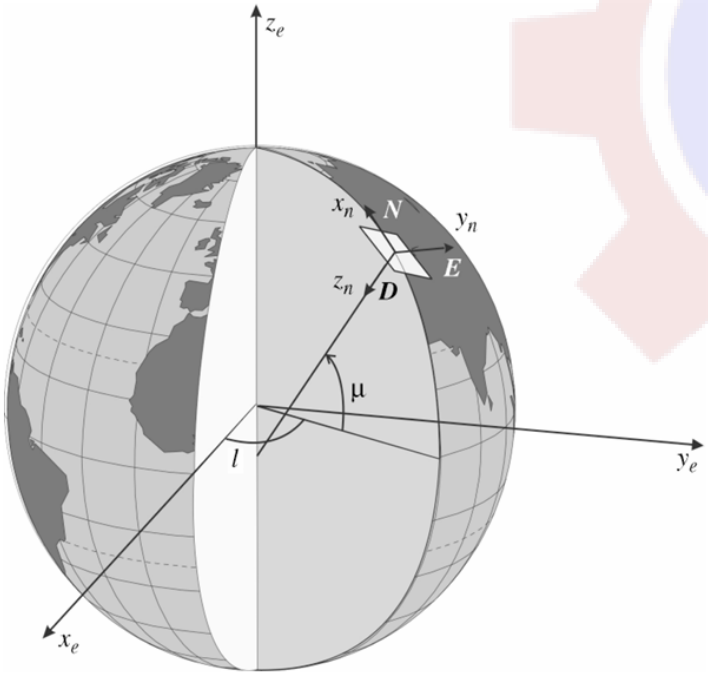
3 DOFs kinematic equations

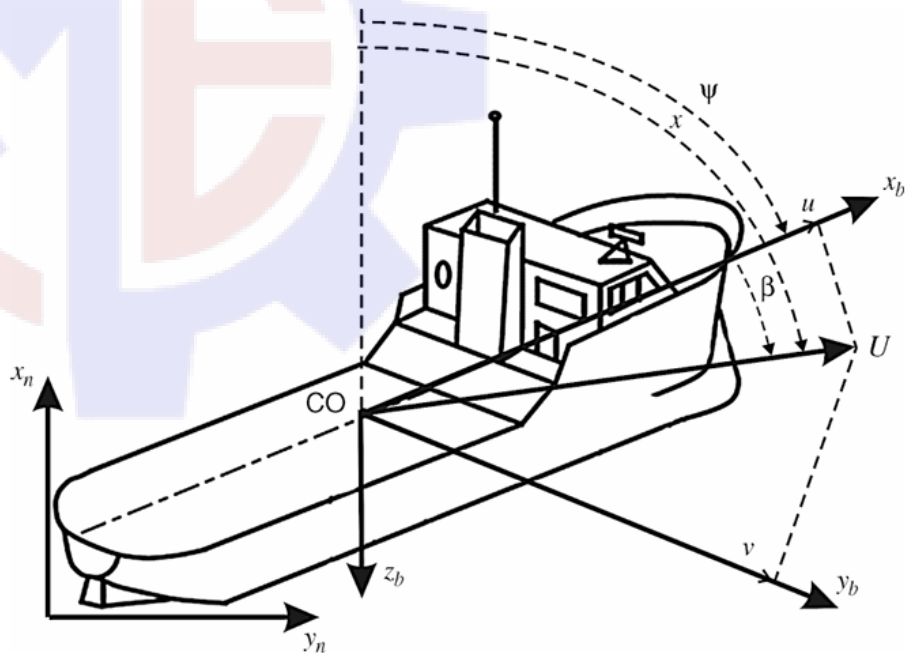
$$\dot{\eta} = R_z(\psi)\nu \tag{3}$$

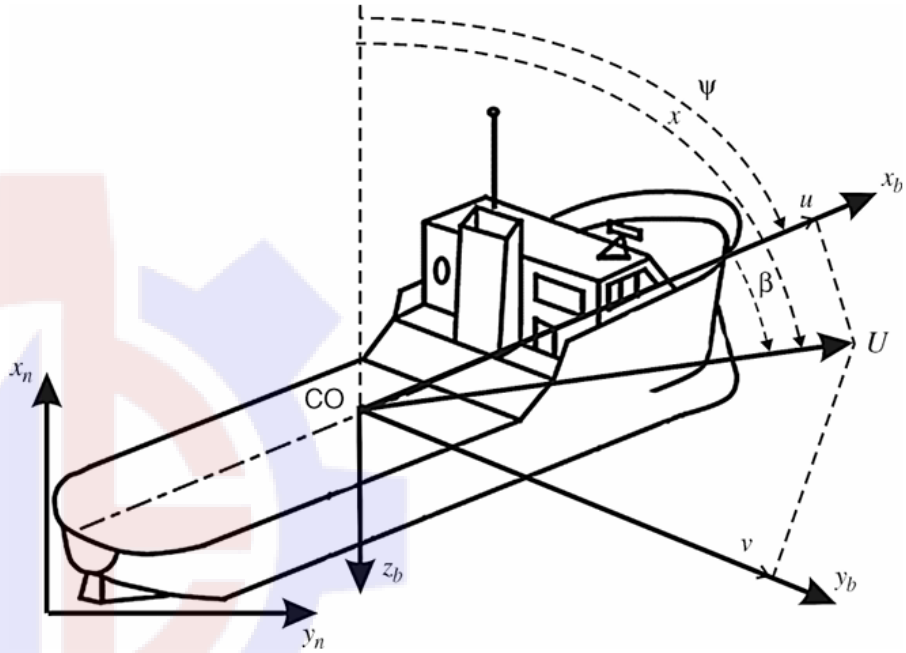
with $\eta = \begin{bmatrix} x & y & \psi \end{bmatrix}^\top$ and $\nu = \begin{bmatrix} u & v & r \end{bmatrix}^\top$

In addition, for more purposes, other transformations are needed

- Transformation between ECEF and NED: wide area or terrestrial guidance and navigation.
- Transformation between BODY and FLOW: express the hydrodynamic data





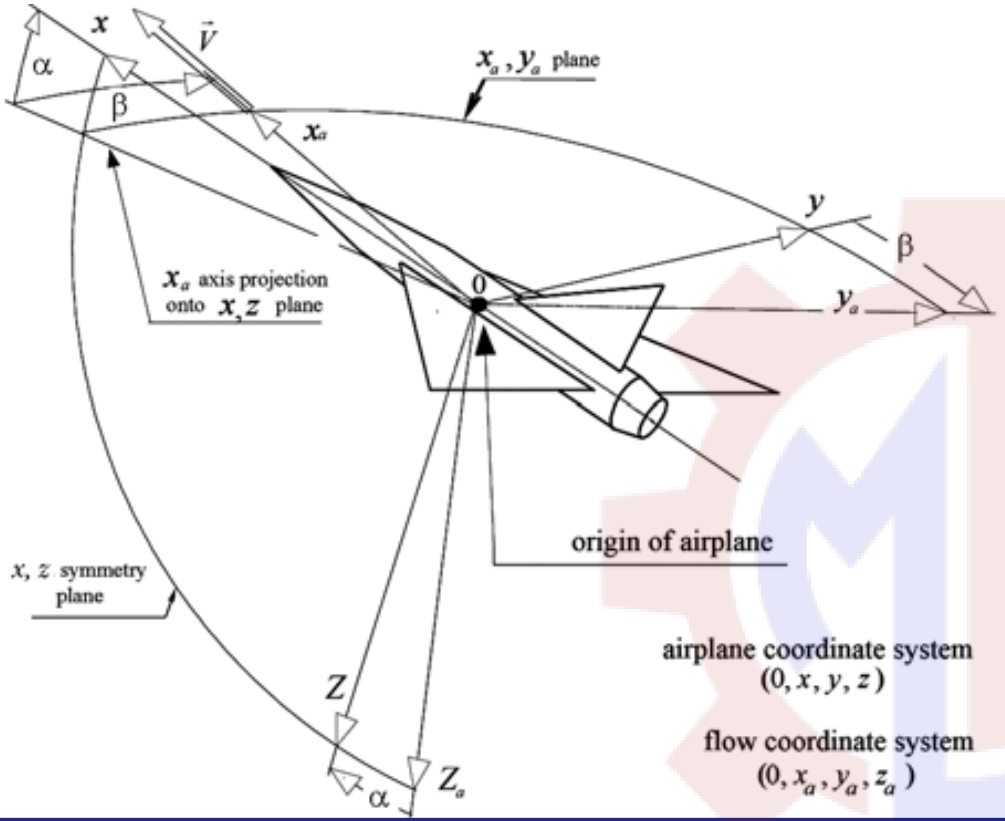


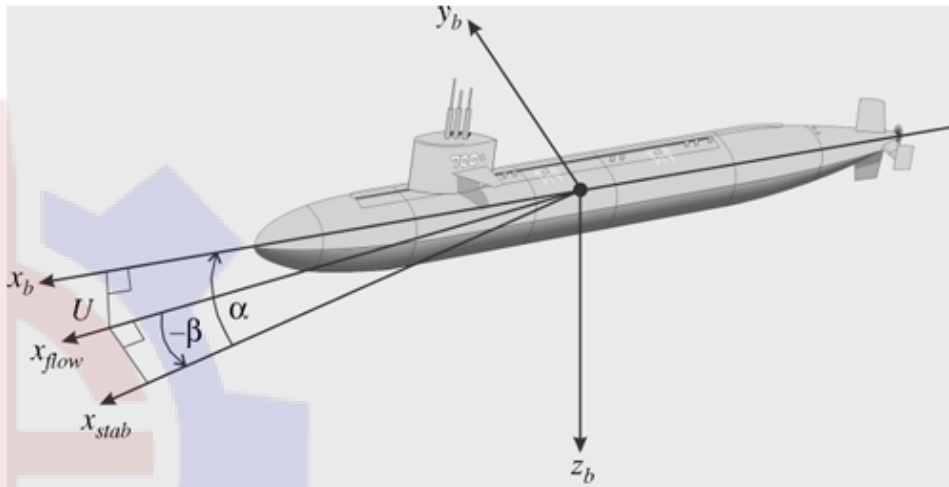
Similarly, in the first case, the craft is assumed to be moving on the horizon.

Definition of Course, Heading, and Sideslip Angles

- Course angle χ: The rotation of x_n and **velocity vector** u of craft around z_n
- Heading angle ψ: Yaw angle of craft
- Sideslip (Drift) angle β: The rotation of x_b and **velocity vector** u of craft around z_b

Thus $\chi = \psi + \beta$.





BODY to FLOW transformation

$$R_b^{flow} = R_{z,-\beta} R_{y,\alpha} = \begin{bmatrix} \cos\beta\cos\alpha & \sin\beta & \cos\beta\sin\alpha \\ -\sin\beta\cos\alpha & \cos\beta & -\sin\beta\sin\alpha \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{4}$$

Table of Contents

Final goal

To find the equations that describe the behavior of the system

$$M_{RB}\dot{\nu} + C_{RB}\nu = \tau_{RB} \tag{5}$$

in addition, as discussed above,

$$\dot{\eta} = J_{\Theta}(\eta)\nu \tag{6}$$

- Newton-Euler equations of motion about CG (Center gravity)
- Newton-Euler equations of motion about CO

Newton's second law

The Newton-Euler formulation is based on Newton's second law, which relates mass m, acceleration $\dot{\vec{v}}_{g/i}$ and force $\vec{f}_g$

$$m\dot{\vec{v}}_{g/i} = \vec{f}_g \tag{7}$$

Euler's First and Second Axioms

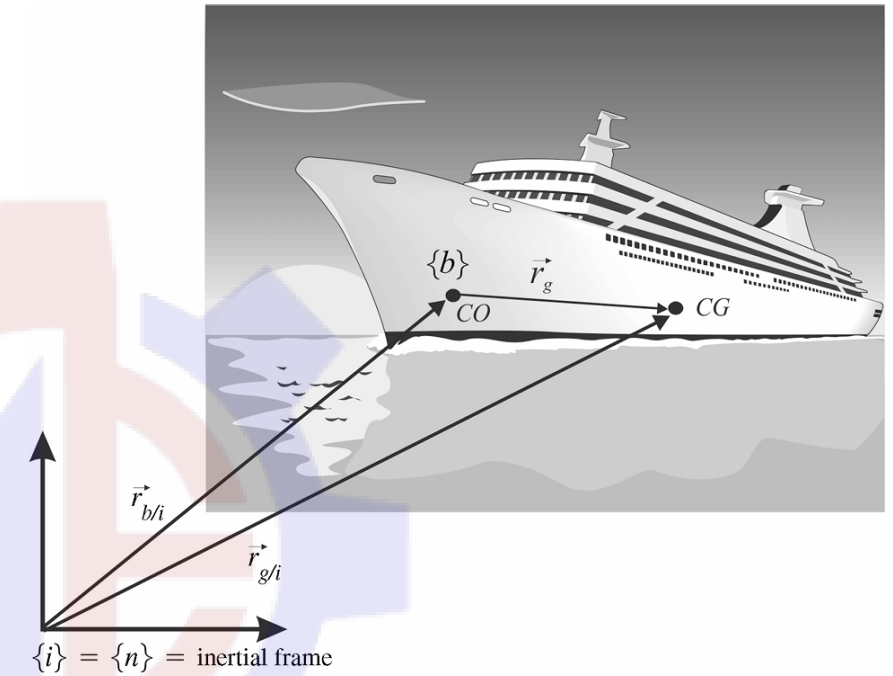
The relationship between linear momentum and angular momentum

$$\frac{d}{dt}\vec{p}_g = \vec{f}_g \tag{8}$$

$$\frac{d}{dt}\vec{h}_g = \vec{m}_g \tag{9}$$

where $\vec{p}_g$ and $\vec{h}_g$ are the linear and angular momentum, respectively. $\vec{f}_g$ and $\vec{m}_g$ are the forces and moments acting on the CG.

- Translational motion
- Rotational motion



Equations of Motion about CG

The Newton-Euler equation could be represented in the matrix form according to

$$M_{RB}^{CG} \begin{bmatrix} \dot{v}_{g/n}^{b} \\ \dot{\omega}_{b/n}^{b} \end{bmatrix} + C_{RB}^{CG} \begin{bmatrix} v_{g/n}^{b} \\ \omega_{b/n}^{b} \end{bmatrix} = \begin{bmatrix} f_g^b \\ m_g^b \end{bmatrix} \tag{10}$$

Translational motion

Using Euler's first axiom,

$$f_g = \frac{d}{dt}(m v_{g/i}) \tag{11}$$

by several tedious mathematical transformations

$$m\left(\dot{v}^b_{g/n} + \omega^b_{b/n} \times v^b_{g/n}\right) = f^b_g \tag{12}$$

Rotational motion

Using Euler's second axiom,

$$m_g = \frac{d}{dt}(I_g \omega_{b/i}) \tag{13}$$

by several tedious mathematical transformations

$$I\dot{\omega}^b_{b/n} - S(I_g \omega^b_{b/n}) \times \omega^b_{b/n} = m^b_g \tag{14}$$

By several SUPER TEDIOUS mathematical transformations

Nonlinear 6 DOF Rigid-Body Equations of Motion

$$m\left[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})\right] = X, \tag{15}$$

$$m\left[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})\right] = Y, \tag{16}$$

$$m\left[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(r\dot{p})\right] = Z, \tag{17}$$

$$\begin{aligned} &I_x\dot{p} + (I_z - I_y)qr - (r + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\ &\quad + m\left[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)\right) = K, \end{aligned} \tag{18}$$

$$\begin{aligned} &I_y\dot{q} + (I_x - I_z)rp - (p + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \\ &\quad + m\left[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)\right) = M, \end{aligned} \tag{19}$$

$$\begin{aligned} &I_z\dot{r} + (I_y - I_x)pq - (q + rp)I_{xz} + (q^2 - p^2)I_{xx} + (rq - \dot{p})I_{zx} \\ &\quad + m\left[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)\right) = N. \end{aligned} \tag{20}$$

Rigid-body dynamics model

$$M_{RB}\dot{\nu} + C_{RB}\nu = \tau_{RB} \tag{21}$$

where

$$M_{RB} = \begin{bmatrix} mI_{3\times 3} & -mS(r_g^b) \\ mS(r_g^b) & I_b \end{bmatrix} \tag{22}$$

$$C_{RB} = \begin{bmatrix} 0_{3\times 3} & -S(M_{11}\nu_1 + M_{12}\nu_2) \\ -S(M_{11}\nu_1 + M_{12}\nu_2) & -S(M_{21}\nu_1 + M_{22}\nu_2) \end{bmatrix} \tag{23}$$

- The Origin CO coincides with the CG
- Translation of the origin CO such that I_b becomes diagonal

Simplifyed 6 DOF Rigid body Equation of Motion

$$m\left[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})\right] = X, \tag{24}$$

$$m\left[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})\right] = Y, \tag{25}$$

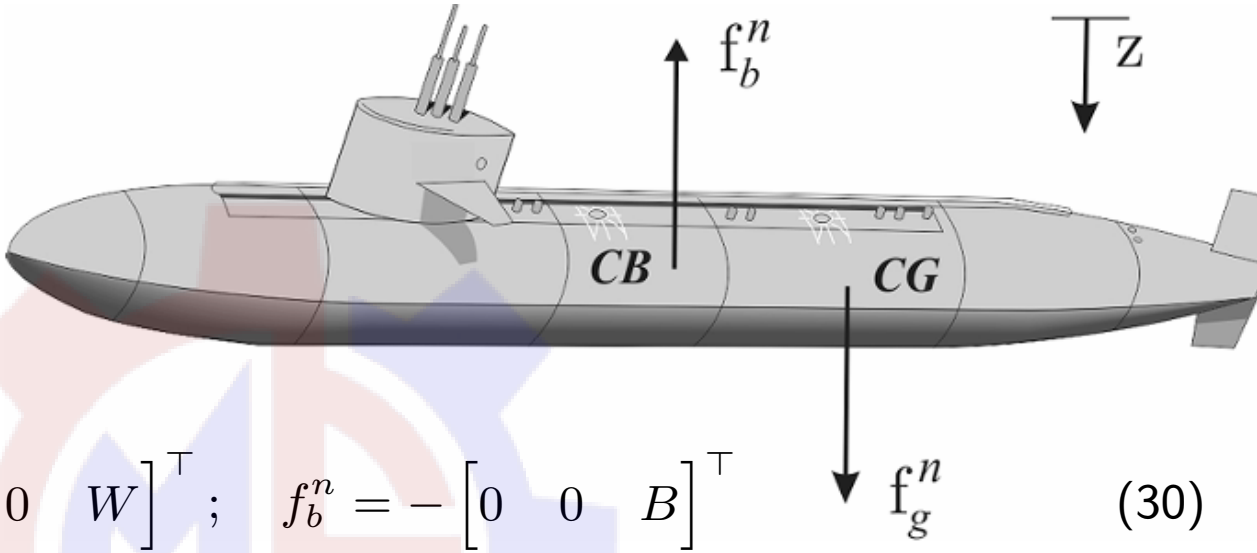
$$m\left[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})\right] = Z, \tag{26}$$

$$I_x\dot{p} + (I_z - I_y)qr + m\left[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)\right] = K, \tag{27}$$

$$I_y\dot{q} + (I_x - I_z)rp + m\left[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)\right] = M, \tag{28}$$

$$I_z\dot{r} + (I_y - I_x)pq + m\left[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)\right] = N. \tag{29}$$

Table of Contents



Forces act in the vertical plane

$$f_g^n = \begin{bmatrix} 0 & 0 & W \end{bmatrix}^\top; \quad f_b^n = -\begin{bmatrix} 0 & 0 & B \end{bmatrix}^\top \tag{30}$$

Thus, the gravity vector is

$$g(\eta) = -\begin{bmatrix} f_g^b + f_b^b \\ r_g^b \times f_g^b + r_b^b \times f_b^b \end{bmatrix} \tag{31}$$

$$= -\begin{bmatrix} R_b^n(\Theta_{nb})^{-1}\left(f_g^n + f_b^n\right) \\ r_g^b \times R_b^n(\Theta_{nb})^{-1} f_g^n + r_b^b \times R_b^n(\Theta_{nb})^{-1} f_b^n \end{bmatrix}. \tag{32}$$

Note: These formulations become more complex when considering the surface vessels

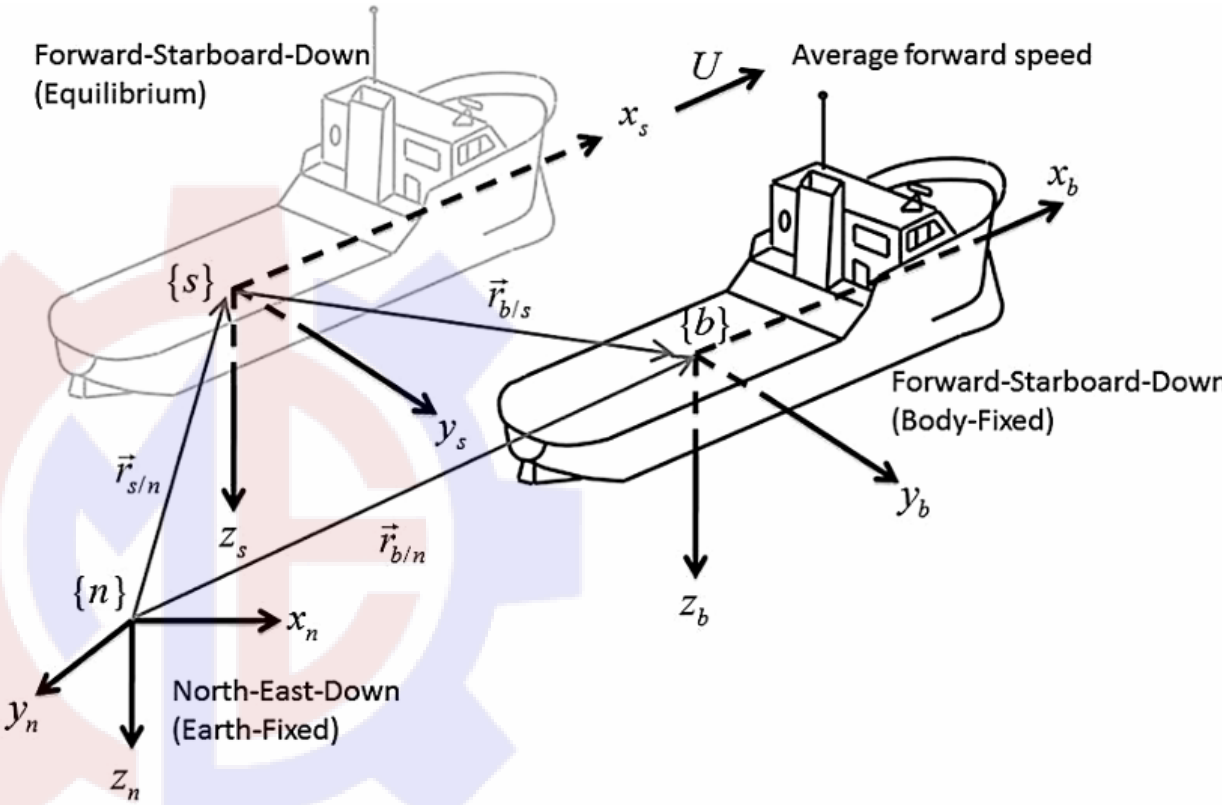
Vehicle-Ocean dynamics model

$$\dot{\eta} = J_\Theta(\eta)\nu$$

$$M_{RB}\dot{\nu} + C^*_{RB}\nu + M_A\dot{\nu}_r + C^*_A\nu_r + (D_P + D_V)\nu_r + \mu + G\eta + g_o = \tau_{\text{wind}} + \tau_{\text{wave}} + \tau$$

Vehicle-Ocean dynamics model

$$\begin{aligned}
\text{Inertia forces:} \quad & M_{RB}\dot{\nu} + C^*_{RB}\nu + M_A\dot{\nu}_r + C^*_A\nu_r \\
\text{Damping forces:} \quad & + (D_P + D_V)\nu_r + \mu \\
\text{Restoring forces:} \quad & + G\eta + g_o \\
\text{Wind and wave forces:} \quad & = \tau_{\text{wind}} + \tau_{\text{wave}} \\
\text{Propulsion forces:} \quad & + \tau
\end{aligned}$$



Several notations

- $v^n_{s/n} = \begin{bmatrix} U\cos\bar{\psi} & U\sin\bar{\psi} & 0 \end{bmatrix}^\top$
- $\omega^n_{s/n} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$
- $\Theta_{ns} = \begin{bmatrix} 0 & 0 & \bar{\psi} \end{bmatrix}^\top$

Define the perturbation coordinate

$$\delta\eta = \begin{bmatrix} r^s_{b/s} \\ \Theta_{sb} \end{bmatrix}; \quad \delta\nu = \begin{bmatrix} v^b_{b/s} \\ \omega^b_{b/s} \end{bmatrix}; \quad \xi = \delta\eta; \quad \Theta_{sb} = \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} \tag{33}$$

Thus, the following equations are achieved as the transformation between BODY and SEAKEEPING coordinates

Transformation between BODY and SEAKEEPING

$$\dot{\nu} = -UL\delta\nu + \delta\dot{\nu} \tag{34}$$

where U is the velocity of the vehicle and $L \in \mathbb{R}_{6\times 6}$

Linear transformations

$$\delta\nu \approx \nu + U(L\delta\eta - e_1) \tag{35}$$

$$\delta\dot{\nu} \approx \dot{\nu} + UL\nu \tag{36}$$

From Cummins Equation in SEAKEEPING Coordinates, the following equations are introduced

Frequency-domain seakeeping equation of motion

$$\Big(-\omega^2(M_{RB} + \bar{A}(\omega)) - j\omega B(\omega) + C \Big)\ddot{\xi}(j\omega) = \tau_{wind}(j\omega) + \tau_{wave}(j\omega) + \delta\tau(j\omega) \quad (37)$$

Time-domain seakeeping equation of motion

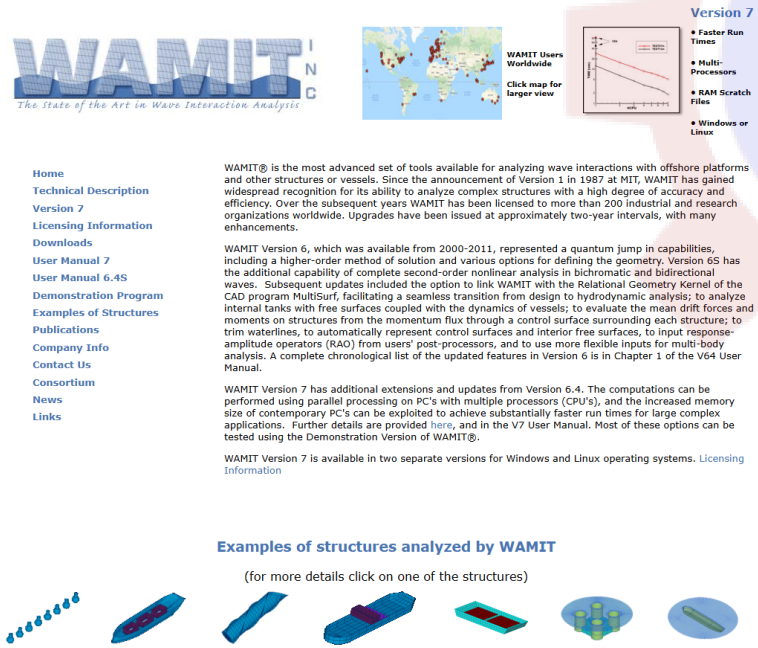
$$\Big(M_{RB} + \bar{A}\Big)\ddot{\xi} + \int_{-\infty}^{t} \bar{K}(t-\tau)\dot{\xi}(t)d\tau + \bar{C}\xi = \tau_{wind} + \tau_{wave} + \delta\tau \quad (38)$$

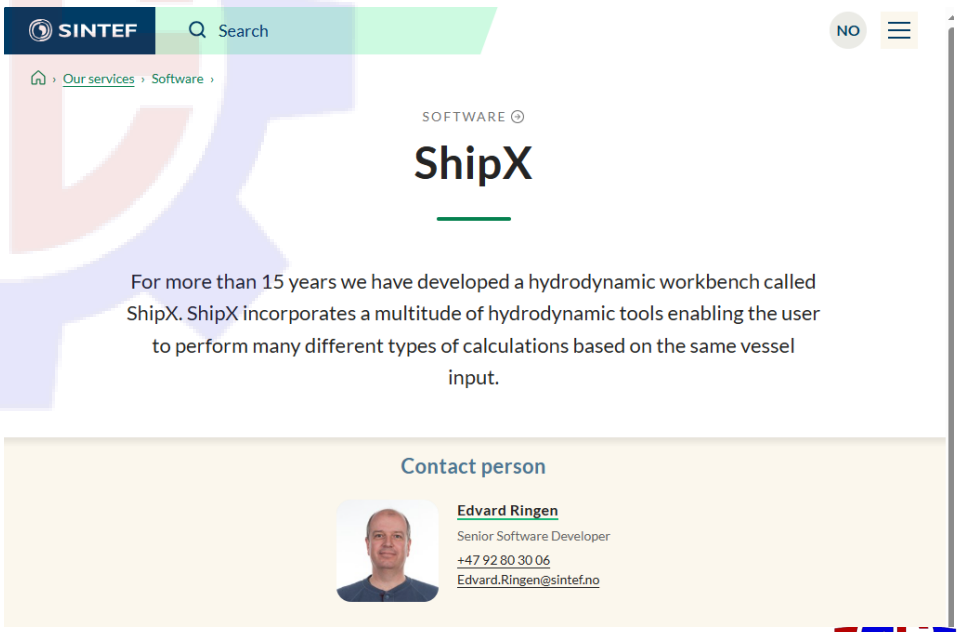
where $\bar{K} = \frac{2}{\pi}\int_0^\infty B(\omega)\cos(\omega t)d\omega$

Notice: For underwater vehicle, $A(\omega) = \text{const}$ and $B(\omega) = 0$

Hydrodynamic added mass forces and moments in 6 DOFs

- The expressions are complicated and not suited for control design.
- Hydrodynamic software programs such as WAMIT and ShipX can be used to compute the added mass term.





From Cummins Equation in SEAKEEPING Coordinates, the following equations are introduced

Linear equation of motion (Zero-Speed Potential Coefficients)

$$M\dot{\nu} + C^*_{RB}\nu + C^*_A\nu_r + D\nu_r + \int_0^t K(t-\tau)[\nu(\tau) - Ue_1]d\tau + G\eta = \sum \tau \tag{39}$$

Linear equation of motion (Speed-Dependent Potential Coefficients)

$$M_U\dot{\nu} + C^*_{RB}\nu + C^*_A\nu_r + D_U\nu_r + \int_0^t K_U(t-\tau)[\nu(\tau) - Ue_1]d\tau + G\eta = \sum \tau \tag{40}$$

where $\sum \tau = \tau_{wind} + \tau_{wind} + \tau$

Goal

$$\underbrace{M_{RB}\dot{\nu} + C_{RB}(\nu)\nu}_{\text{rigid-body forces}} + \underbrace{M_A\dot{\nu}_r + C_A(\nu_r)\nu_r + D(\nu_r)\nu_r}_{\text{hydrodynamic forces}} + \underbrace{g(\eta) + g_o}_{\text{hydrostatic forces}} = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \tag{41}$$

or

$$M\dot{\nu}_r + C(\nu_r)\nu_r + D(\nu_r)\nu_r + g(\eta) + g_o = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \tag{42}$$

where $\nu_r = \nu - \nu_c$ is the relative velocity vector.

- For slender bodies, such as AUV, the longitudinal (surge, heave, and pitch motion) and lateral (sway, roll, and yaw motion) models are referred to consider.
- For ROVs operating, the horizontal-plane (surge, sway, and yaw motion) and depth (heavy) models are referred to consider.

3DOF maneuvering model

The horizontal motions (surge, sway, and heave) of a marine craft moving at forward speed could be described by a zero frequency model, where

$$M_A = \begin{bmatrix} A_{11}(0) & 0 & 0 \\ 0 & A_{22}(0) & A_{26}(0) \\ 0 & A_{62}(0) & A_{66}(0) \end{bmatrix} \tag{43}$$

$$D_p = 0 \tag{44}$$

are the constant matrices.

Limitation One limitation of the zero-frequency assumption is that it cannot be applied to heave, roll, and pitch. For 2nd-order mass-damper-spring systems the dominating frequencies are the natural frequencies.

Extension to 6-DOF models

Solution: Formulate frequency-independent models in heave, pitch, and roll at their respective natural frequencies and not the zero frequency.

$$\omega_3 = \sqrt{\frac{C_{33}}{m + A_{33}(\omega_3)}} \tag{45}$$

$$\omega_4 = \sqrt{\frac{C_{44}}{I_x + A_{44}(\omega_4)}} \tag{46}$$

$$\omega_5 = \sqrt{\frac{C_{55}}{I_y + A_{55}(\omega_5)}} \tag{47}$$

Key assumption: No coupling between the surge-sway-yaw and the heave-roll-pitch subsystems.

The expression for the fluid kinetic energy T_A, see Ref. [3], can be written as a quadratic form of the body axis velocity vector components, that is:

$$T_A = \frac{1}{2}\boldsymbol{\nu}^T M_A \boldsymbol{\nu} \tag{48}$$

Here M_A is a 6×6 added inertia matrix defined as:

$$M_A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \triangleq -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix}, \quad Y_A = Y_{\dot{u}}\dot{u}, \quad Y_{\dot{u}} \triangleq \frac{\partial Y}{\partial \dot{u}} \tag{49}$$

Notice: In a real fluid, these 36 constants could be distinct. With ideal (without friction),$M_A = M_A^\top$

[3] Imlay, Frederick H. *The complete expressions for added mass of a rigid body moving in an ideal fluid*. No. Report 1528. 1961.

Hydrodynamic Coriolis-Centripetal Matrix $C_A(\nu)$

For a rigid body moving through an ideal fluid the hydrodynamic Coriolis and centripetal matrix $C_A(\nu)$ could always be parametrized such that it is skew-symmetric

$$C_A(\nu) = -C_A(\nu)^\top, \quad \forall \nu \tag{50}$$

As mentioned above,

$$C_A(\nu) = \begin{bmatrix} 0 & -S(A_{11}\nu_1 + A_{12}\nu_2) \\ -S(A_{11}\nu_1 + A_{12}\nu_2) & -S(A_{21}\nu_1 + A_{22}\nu_2) \end{bmatrix} \tag{51}$$

Vehicle-Ocean dynamics model

$$\dot{\eta} = J_\Theta(\eta)\nu$$

$$M_{RB}\dot{\nu} + C^*_{RB}\nu + M_A\dot{\nu}_r + C^*_A\nu_r + (D_P + D_V)\nu_r + \mu + G\eta + g_o = \tau_{\text{wind}} + \tau_{\text{wave}} + \tau$$

Vehicle-Ocean dynamics model

$$\begin{aligned}
\text{Inertia forces:} \quad & M_{RB}\dot{\nu} + C^*_{RB}\nu + M_A\dot{\nu}_r + C^*_A\nu_r \\
\text{Damping forces:} \quad & + (D_P + D_V)\nu_r + \mu \\
\text{Restoring forces:} \quad & + G\eta + g_o \\
\text{Wind and wave forces:} \quad & = \tau_{\text{wind}} + \tau_{\text{wave}} \\
\text{Propulsion forces:} \quad & + \tau
\end{aligned}$$

In practical implementations, it is difficult to determine higher-order terms as well as the off-diagonal terms in general expression for hydrodynamic damping.
The different damping terms contribute to both linear and quadratic damping. However, it is in general difficult to separate these effects. In many cases, it is convenient to write total hydrodynamic damping as

$$D(\nu_r) = D + D_n(\nu_r) \tag{52}$$

- Linear viscous damping

$$D = D_p + D_v \approx -diag(X_u, Y_v, Z_w, K_p, M_q, N_r) \tag{53}$$

- Nonlinear surge damping
  - Nonlinear Surge Damping: Damping Due to Vortex Shedding
  - Cross-Flow Drag Principle (lifting Forces:): For relative current angles, the cross-flow drag principle may be applied to calculate the nonlinear damping force in sway and the yaw moment

The approximate diagonal matrix D_p is expressed as:

$$D_p \approx \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & B_{33}(\omega_{\text{heave}}) & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{44}(\omega_{\text{roll}}) & 0 & 0 \\ 0 & 0 & 0 & 0 & B_{55}(\omega_{\text{pitch}}) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{54}$$

The natural frequencies ω_{heave}, ω_{roll}, and ω_{pitch} can be computed as introduced in the above.

A diagonal matrix usually approximates the linear viscous damping terms:

$$D_v \approx \text{diag}\{B_{11v}, B_{22v}, B_{33v}, B_{44v}, B_{55v}, B_{66v}\} \tag{55}$$

where the elements B_{iiv} $(i = 1, \ldots, 6)$ can be computed from the time constants and natural periods of the system (see Section 6.4).

The expressions for the diagonal elements of D_v are as follows:

$$B_{11v} = \frac{m + A_{11}(0)}{T_{\text{surge}}} = \frac{8\pi\zeta_{\text{surge}}[m + A_{11}(0)]}{T_{n,\text{surge}}} \tag{56}$$

$$B_{22v} = \frac{m + A_{22}(0)}{T_{\text{sway}}} = \frac{8\pi\zeta_{\text{sway}}[m + A_{22}(0)]}{T_{n,\text{sway}}} \tag{57}$$

$$B_{33v} = 2\Delta\zeta_{\text{heave}}\omega_{\text{heave}}[m + A_{33}(\omega_{\text{heave}})] \tag{58}$$

$$B_{44v} = 2\Delta\zeta_{\text{roll}}\omega_{\text{roll}}[I_x + A_{44}(\omega_{\text{roll}})] \tag{59}$$

$$B_{55v} = 2\Delta\zeta_{\text{pitch}}\omega_{\text{pitch}}[I_y + A_{55}(\omega_{\text{pitch}})] \tag{60}$$

$$B_{66v} = \frac{I_z + A_{66}(0)}{T_{\text{yaw}}} = \frac{8\pi\zeta_{\text{yaw}}[I_z + A_{66}(0)]}{T_{n,\text{yaw}}} \tag{61}$$

Quadratic drag and lifting in 6DOF

Quadratic drag and lifting in 6DOF

$$D_n(\nu_r)\nu_r = [|\nu_r|^\top D_i(\rho, A, C_D)\nu_r)]_{6\times 1} \tag{62}$$

The viscous damping force due to vortex shedding can be modeled as

$$f(U) = -\frac{1}{2}\rho C_D A|U_r|U_r \tag{63}$$

where ρ is the fluid density, C_D is drag coefficient, A is the projected cross-sectional area, and U is the velocity of the vehicle.

Hydrodynamic Mass–Damper

- **Added mass** M_A due to the inertia of the surrounding fluid (see Section 6.2). The corresponding Coriolis and centripetal matrix due to added mass is due to the rotation of $\{b\}$ with respect to $\{n\}$ and is denoted $C_A(\upsilon_r)$ (see Section 6.3).
- **Radiation-induced potential damping** D_p due to the energy carried away by generated surface waves.
- **Viscous damping** caused by skin friction, wave drift damping, vortex shedding, and lift/drag (see Section 6.4).

The resulting hydrodynamic force is written as:

$$\tau_{\text{hyd}} = -M_A\dot{\nu}_r - C_A(\nu_r)\nu_r - D_p\nu_r + \tau_{\text{visc}} \tag{64}$$

where $\nu_r = \nu - \nu_c$ with $\nu_c = [u_c, v_c, w_c, 0, 0, 0]^\top$ is the relative velocity due to an *irrotational constant ocean current* (see Section 8.3), and

$$\tau_{\text{visc}} = -D_v\upsilon_r - D_n(\upsilon_r)\upsilon_r \tag{65}$$

Hydrostatic Spring Stiffness: **Restoring forces** due to Archimedes

$$\tau_{\text{hs}} = -g(\eta) - g_o \tag{66}$$

Table of Contents

Environmental Disturbances

The environmental disturbances include

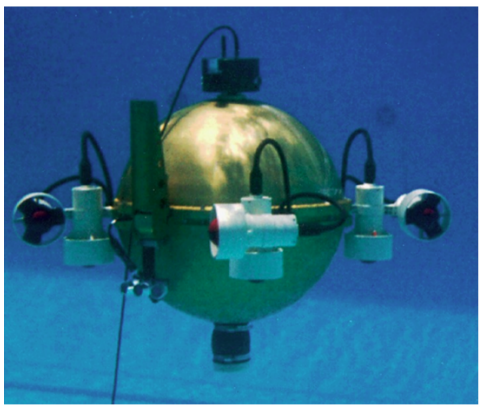
- Wind forces and moments
- Wave forces and moments

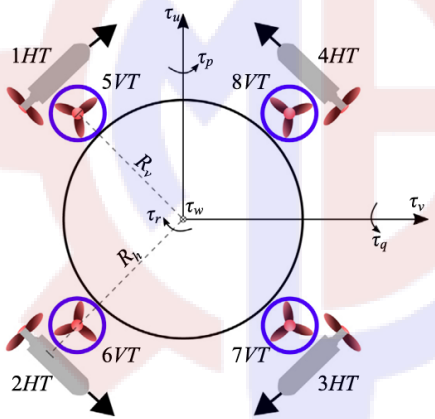
In this context, if the autonomous underwater vehicles are considered to be working at the deep water level, the wind and wave disturbances are not significant enough to be considered.

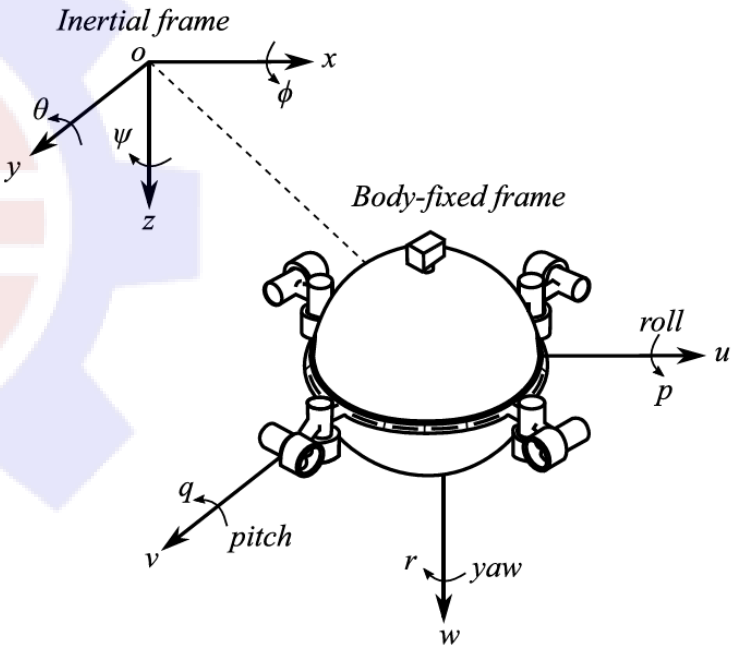
The wind and wave disturbances become more significant when we consider a surface vehicle, and it could be introduced in the future version.
[Coming soon!!]

Table of Contents

ODIN = Omni-directional intelligent navigation [4, 5]







[4] Choi, Hyun-Taek, et al. "Development of an underwater robot, ODIN-III." Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453). Vol. 1. IEEE, 2003.

[5] Ali, Nihad, Isaac Tawiah, and Weidong Zhang. "Finite-time extended state observer based nonsingular fast terminal sliding mode control of autonomous underwater vehicles." Ocean Engineering 218 (2020): 108179.

Dynamic equations

$$M_{RB}\dot{\nu} + C_{RB}\nu = \tau_{RB} \tag{67}$$

where

$$M_{RB} = \begin{bmatrix} mI_{3\times 3} & -mS(r_g^b) \\ mS(r_g^b) & I_b \end{bmatrix}; C_{RB} = \begin{bmatrix} mS(\nu_2) & -mS(\nu_2)S(r_g^b) \\ mS(\nu_2)S(r_g^b) & -S(I_b\nu_2) \end{bmatrix}$$

Notice that, the term $C_{RB}(\nu)$ is modified by several mathematical transformations such as $C_{RB}(\nu) = C_{RB}(\nu_r)$ by linear velocity -independent parametrizations. Thus, the above dymanic equation could be represented as

$$M_{RB}\dot{\nu}_r + C_{RB}\nu_r = \tau_{RB} \tag{68}$$

with ν_r is the relative velocity of the vehicle. Therefore, the kinematics equation is represented as

$$\dot{\eta} = J(\eta)\nu \tag{69}$$

where $J(\eta)$ is the Jacobian matrix.

In order to lose the general, let's consider the ODIN as a spherical solid. Thus, the ODIN-geometry could be described as the following equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 0 \tag{70}$$

Let's define [6]

$$\alpha_0 = abc \int_0^\infty \frac{\mathrm{d}\lambda}{(a^2+\lambda)\,\Delta}, \quad \beta_0 = abc \int_0^\infty \frac{\mathrm{d}\lambda}{(b^2+\lambda)\,\Delta}, \quad \gamma_0 = abc \int_0^\infty \frac{\mathrm{d}\lambda}{(c^2+\lambda)\,\Delta} \tag{71}$$

with $\Delta = \left\{(a^2+\lambda)(b^2+\lambda)(c^2+\lambda)\right\}^{\frac{1}{2}}$

Special case: $a = b = c = R$, it is easy to achieve that

$$\alpha_0 = \beta_0 = \gamma_0 = R^3 \int_0^\infty \frac{\mathrm{d}\lambda}{(R^2+\lambda)^2\sqrt{(R^2+\lambda)}} = R^3 \frac{2}{3R^3} = \frac{2}{3} \tag{72}$$

[6] Lamb, Horace. Hydrodynamics. University Press, 1924.

Following [7], the added mass of the spherical-solid underwater vehicle could be represented as

$$X_{\dot{u}} = -\frac{\alpha_0}{2-\alpha_0}\frac{4}{3}\pi\rho abc, \tag{73}$$

$$Y_{\dot{v}} = -\frac{\beta_0}{2-\beta_0}\frac{4}{3}\pi\rho abc, \tag{74}$$

$$Z_{\dot{w}} = -\frac{\gamma_0}{2-\gamma_0}\frac{4}{3}\pi\rho abc, \tag{75}$$

$$K_{\dot{p}} = -\frac{1}{5}\frac{(b^2-c^2)(\gamma_0-\beta_0)}{2(b^2-c^2)+(b^2+c^2)(\beta_0-\gamma_0)}\frac{4}{3}\pi\rho abc, \tag{76}$$

$$M_{\dot{q}} = -\frac{1}{5}\frac{(c^2-a^2)(\alpha_0-\gamma_0)}{2(c^2-a^2)+(c^2+a^2)(\gamma_0-\alpha_0)}\frac{4}{3}\pi\rho abc, \tag{77}$$

$$N_{\dot{r}} = -\frac{1}{5}\frac{(a^2-b^2)(\beta_0-\alpha_0)}{2(a^2-b^2)+(a^2+b^2)(\alpha_0-\beta_0)}\frac{4}{3}\pi\rho abc. \tag{78}$$

[7] Imlay, Frederick H. The complete expressions for added mass of a rigid body moving in an ideal fluid. No. Report 1528. 1961.

Special case: $a = b = c = R$, then $\alpha_0 = \beta_0 = \gamma_0 = \frac{2}{3}$, therefore

$$X_{\dot{u}} = Y_{\dot{v}} = Z_{\dot{w}} = -\rho\frac{2}{3}\pi R^3 \tag{79}$$

$$K_{\dot{p}} = K_{\dot{p}} = K_{\dot{p}} = 0 \tag{80}$$

Written in matrix form,

$$M_A = \begin{bmatrix} \rho\frac{2}{3}\pi R^3 I_3 & 0_3 \\ 0_3 & 0_3 \end{bmatrix} \tag{81}$$

Then, it is easy to find the Coriolis term of added mass, which is not depended on ν_1

$$C_A(\nu_r) = \rho\frac{2}{3}\pi R^3 \begin{bmatrix} 0 & 0 & 0 & 0 & w_r & -v_r \\ 0 & 0 & 0 & -w_r & 0 & u_r \\ 0 & 0 & 0 & v_r & -u_r & 0 \\ 0 & w_r & -v_r & 0 & 0 & 0 \\ -w_r & 0 & u_r & 0 & 0 & 0 \\ v_r & -u_r & 0 & 0 & 0 & 0 \end{bmatrix} \tag{82}$$

Hydrostatics force of Submerged Vehicles is represented as

$$g(\eta) = \begin{bmatrix} (W-B)\sin(\theta) \\ -(W-B)\cos(\theta)\sin(\phi) \\ -(W-B)\cos(\theta)\cos(\phi) \\ -(y_g W - y_b B)\cos(\theta)\cos(\phi) + (z_g W - z_b B)\cos(\theta)\sin(\phi) \\ (z_g W - z_b B)\sin(\theta) + (x_g W - x_b B)\cos(\theta)\cos(\phi) \\ -(x_g W - x_b B)\cos(\theta)\sin(\phi) - (y_g W - y_b B)\sin(\theta) \end{bmatrix} \tag{83}$$

with (x_g, y_g, z_g) and (x_b, y_b, z_b) are the coordinates of CO and CB, respectively.

Special case:

- $W = B \rightarrow$ neutral buoyant, the Gravitational/buoyancy forces just cause the rotations (moments).
- Oz_b and Oz_g are coincidence. That mean $x_g = x_b$ and $y_g = y_b$.
- $CB \equiv CG \equiv CO$ and $W = B \rightarrow g(\eta) = (0, 0, 0, 0, 0, 0)^\top$.

Damping includes

- Linear damping

$$D = -diag(X_u, Y_v, Z_w, K_p, M_q, N_r) \tag{84}$$

- Nonlinear term (important for high-speed vehicles)

$$d(\nu_r) = \begin{bmatrix} \frac{1}{2}V_r^2 S C_D(\alpha)\cos\alpha - \frac{1}{2}V_r^2 S C_L(\alpha)\sin\alpha \\ Y_{cross} \\ -\frac{1}{2}V_r^2 S C_D(\alpha)\sin\alpha - \frac{1}{2}V_r^2 S C_L(\alpha)\cos\alpha \\ 0_{3\times 1} \end{bmatrix} \tag{85}$$

More details, C_D, C_L are the drag, lifting coefficient, respectively.

$$F_{drag} = -\frac{1}{2}\rho V_r^2 S C_D(\alpha) \tag{86}$$

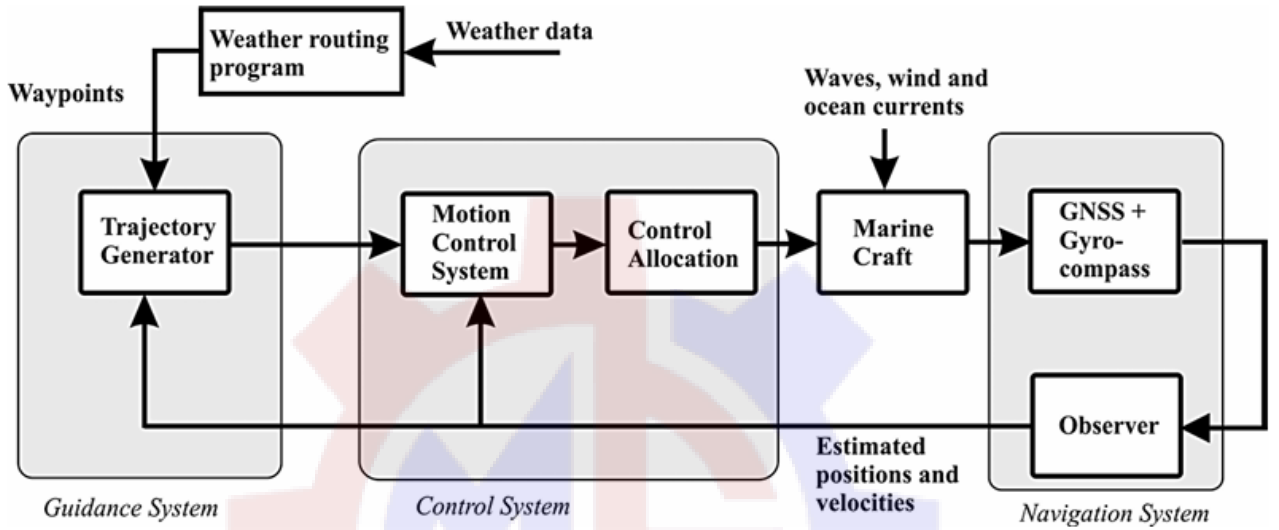
$$F_{lift} = -\frac{1}{2}\rho V_r^2 S C_L(\alpha) \tag{87}$$

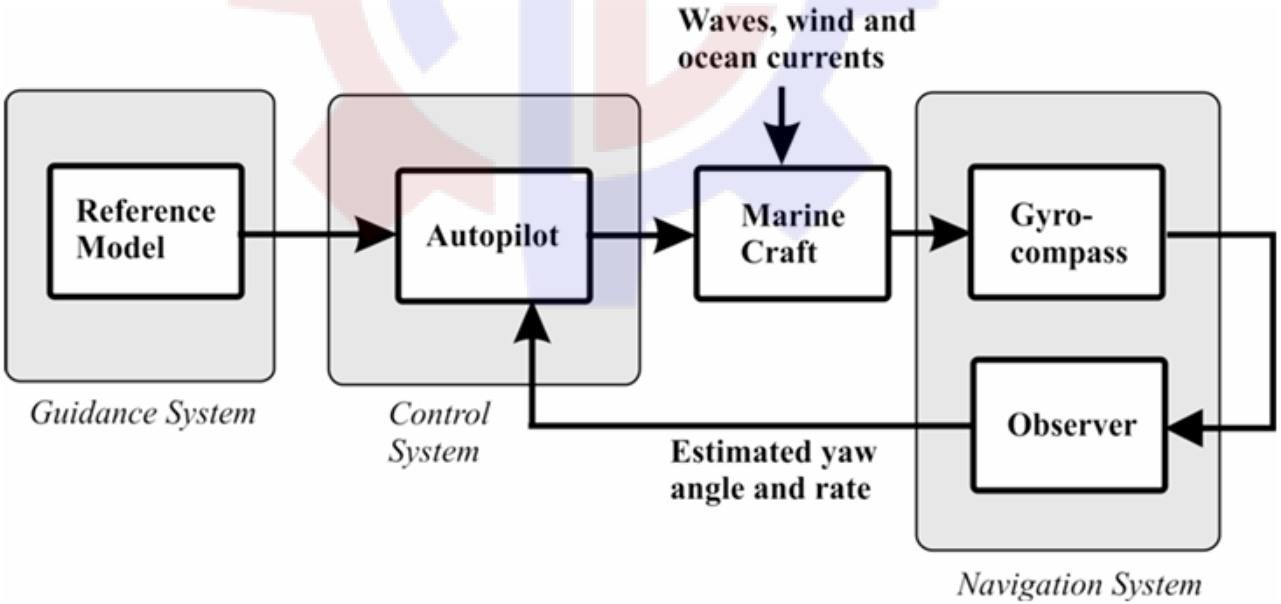
$$Y_{cross} = -\frac{1}{2}\rho \int_{-L/2}^{L/2} T(x) C_d^{2D}(x) |v_r + xr|(v_r + xr)dx \tag{88}$$

Table of Contents

In the previous section, we introduce the modeling and mathematical model to describe the dynamics behavior. In the following steps, the control techniques, which include guidance, navigation and motion control, for a specific underwater vehicle (in general, consider the ocean vehicle). The theory and cases studies are organized as three independent part:

- **G - Guidance Systems**: Systems for automatically guiding the path of a marine craft, usually without direct or continuous human control.
- **N - Navigation Systems**: Systems for determination of the craft's position/attitude, velocity and acceleration.
- **C - Control Systems**: PID design methods for automatic control of position/attitude, velocity and acceleration. This involves control systems for stabilization, trajectory-tracking and path following control of marine craft.

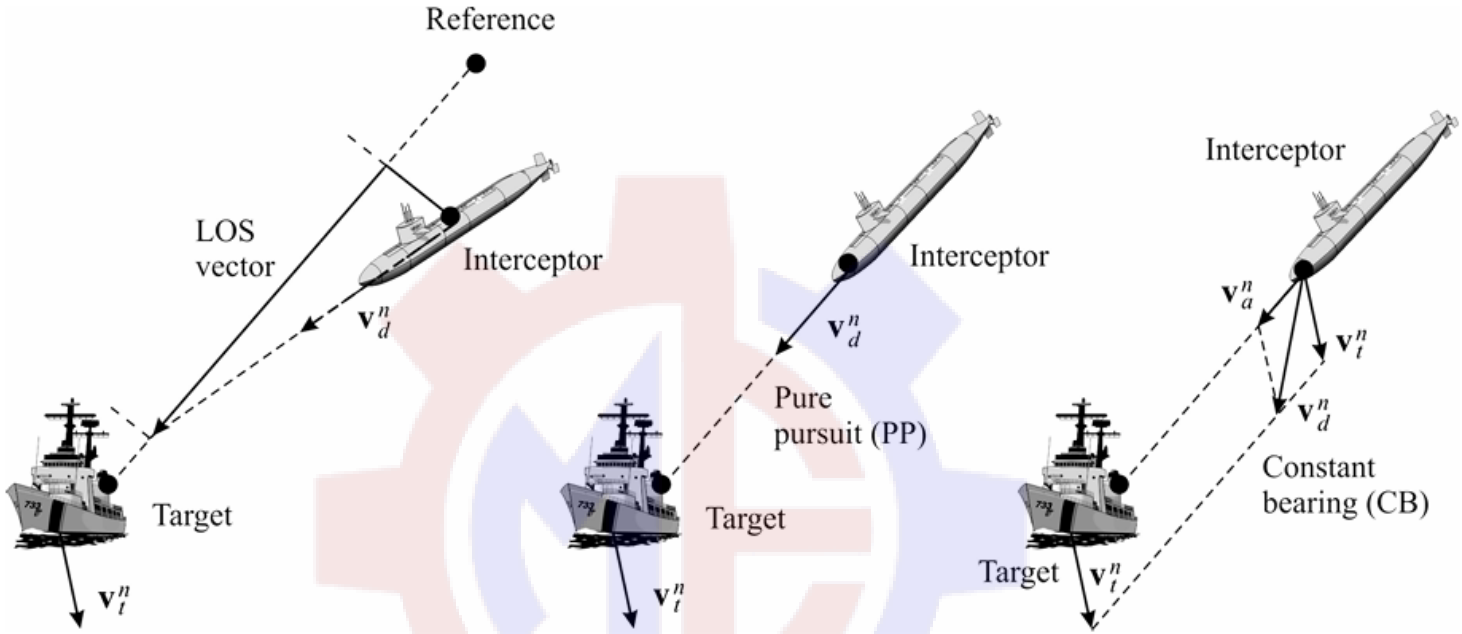

Weather routing program
Weather data
Waypoints
Waves, wind and ocean currents
Trajectory Generator
Motion Control System
Control Allocation
Marine Craft
GNSS + Gyro-compass
Observer
Estimated positions and velocities
Guidance System
Control System
Navigation System


Waves, wind and ocean currents
Reference Model
Autopilot
Marine Craft
Gyro-compass
Observer
Estimated yaw angle and rate
Guidance System
Control System
Navigation System

Setpoint Regulation: The most basic guidance system is a constant input or setpoint provided by a human operator. The corresponding controller will then be a regulator. Examples of setpoint regulation are constant depth, trim, heel and speed control. It could also be regulation to zero, which is commonly required in roll and pitch for instance.

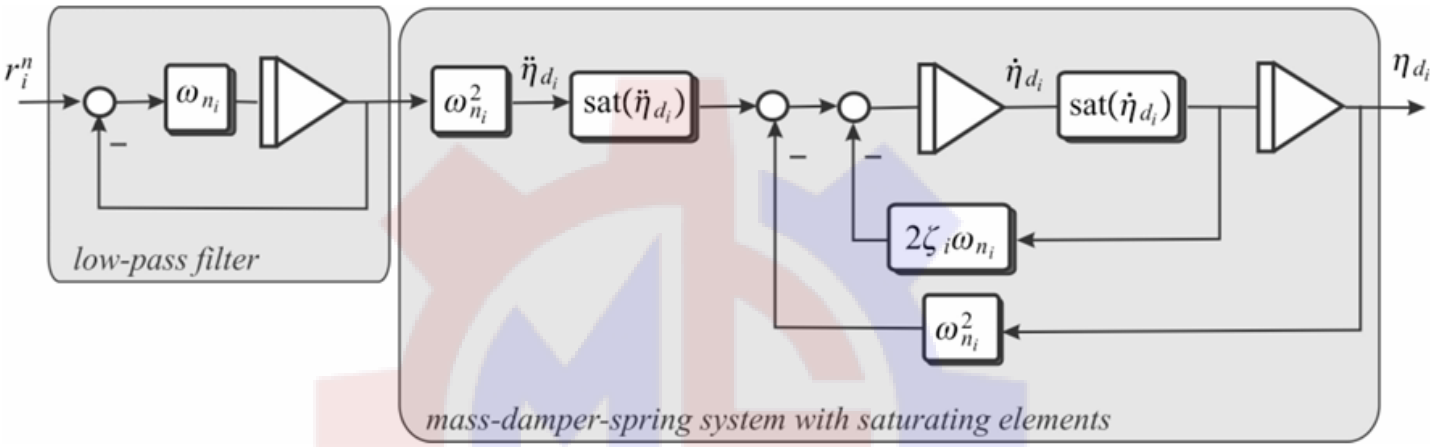
Trajectory-Tracking Control: The position and velocity of the marine craft should track desired time varying position and velocity reference signals. The corresponding feedback controller is a trajectory tracking controller. Tracking control can be used for course-changing maneuvers, speed-changing and attitude control. An advanced guidance system computes optimal time-varying trajectories from a dynamic model for a predefined control objective. If a constant setpoint is used as input to a low-pass filter (reference model) in an open-loop guidance system, the outputs of the filter will be smooth time-varying reference trajectories for position, velocity and acceleration (PVA).

Path-Following Control: This is to follow a predefined path independent of time (no temporal constraints). Moreover, no restrictions are placed on the temporal propagation along the path. This is typical for ships in transit between continents or underwater vehicles used to map the seabed.



Several guidance strategies

- Line-of-Sight Guidance (LOS)
- Pure Pursuit Guidance (PP)
- Constant Bearing Guidance (CB)
- ...



The easiest approach is that the desired trajectory can be computed using reference models generated by low-pass filters

$$\frac{x_d}{r}(s) = h_{lp}(s) = \frac{\omega^2}{s^2 + 2\xi\omega s + \omega^2} \tag{89}$$

where $r(s)$ is set up by operator and $x_d(s)$ is the controller input.

The second one is that we consider the simulation model to generate the trajectory. For instance, a dynamic model could be chosen as

$$\dot{\eta}_d = J(\eta)\nu_d \tag{90}$$

$$M\dot{\nu}_d + N\nu_d + g(\nu_d) = \tau \tag{91}$$

The smooth trajectory could be generated by simulating the above dynamic model with a certain controller, such as PD control. To explain,

$$\tau = g(\eta_d) - J^{-1}(\eta_d)(K_p(\eta_d - \eta_{ref}) + Kd\dot{\eta}_d) \tag{92}$$

The vehicles' trajectory could be generated by optimization problems. The easiest approach method for this problem by consider a cost function of power, or time, and the purpose is to minimize this cost function. That mean,

$$x_d* = \text{argmin}_{x_d}\Big(f(x,t)\Big) \tag{93}$$

$$s.t. \quad x_d \in \mathcal{X} \tag{94}$$

For examples, by considering the discrete model of vehicles, the optimization problem could be expressed in detail as follows

$$x_d^* = \text{argmin}_{x_d} \sum x_d(k)^\top Q x_d(k) \tag{95}$$

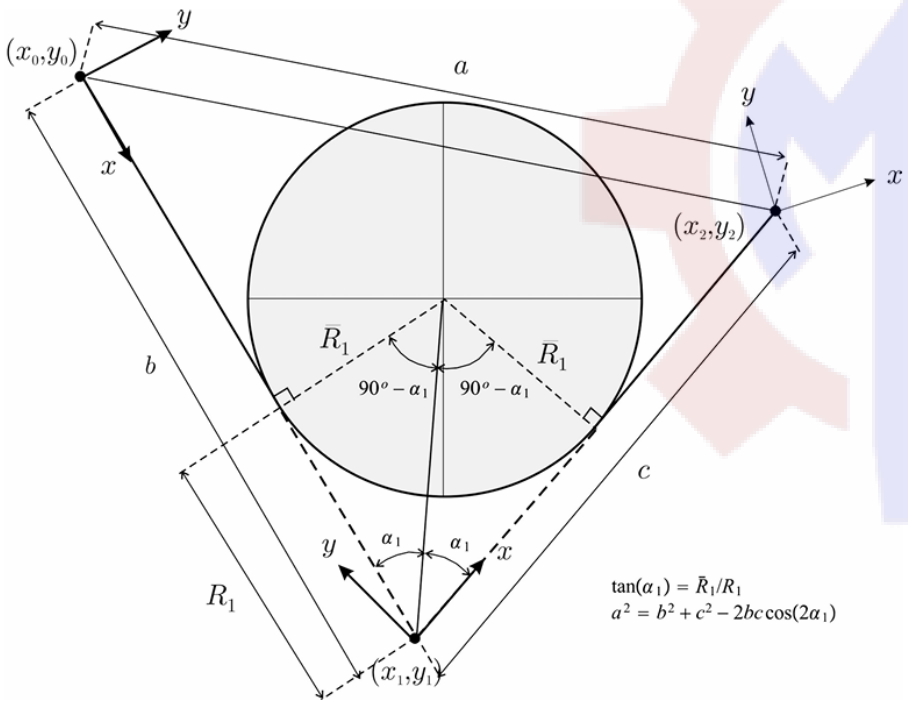
$$or \tag{96}$$

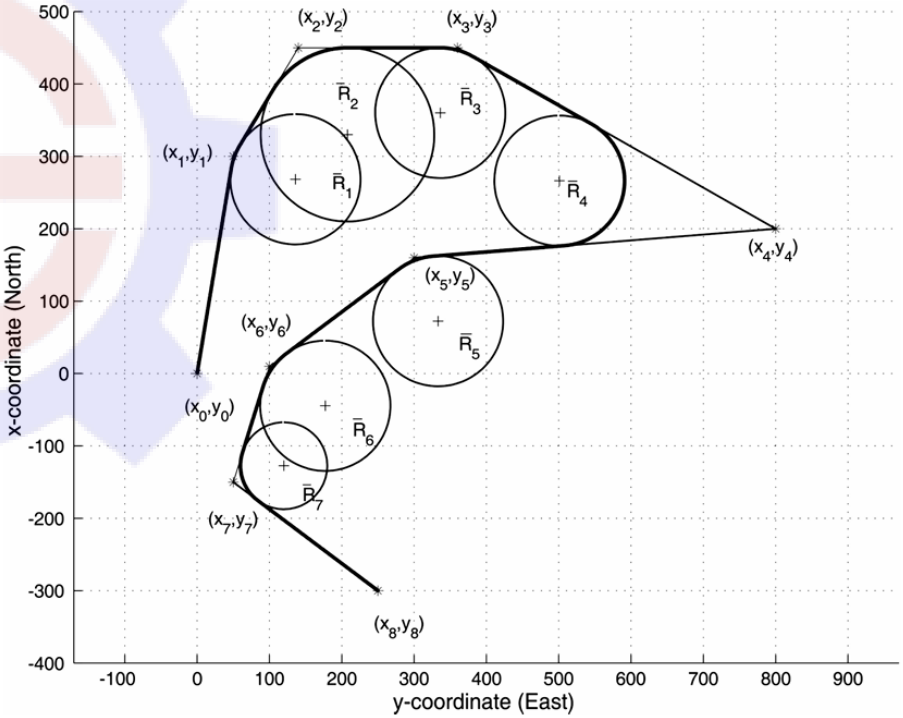
$$x_d^* = \text{argmin}_{x_d} t \tag{97}$$

with constraint $x_d(k+1) = f(x_d(k), u(k))$ and $Q = Q^\top > 0$

Path generation using straight lines and Circular Arcs

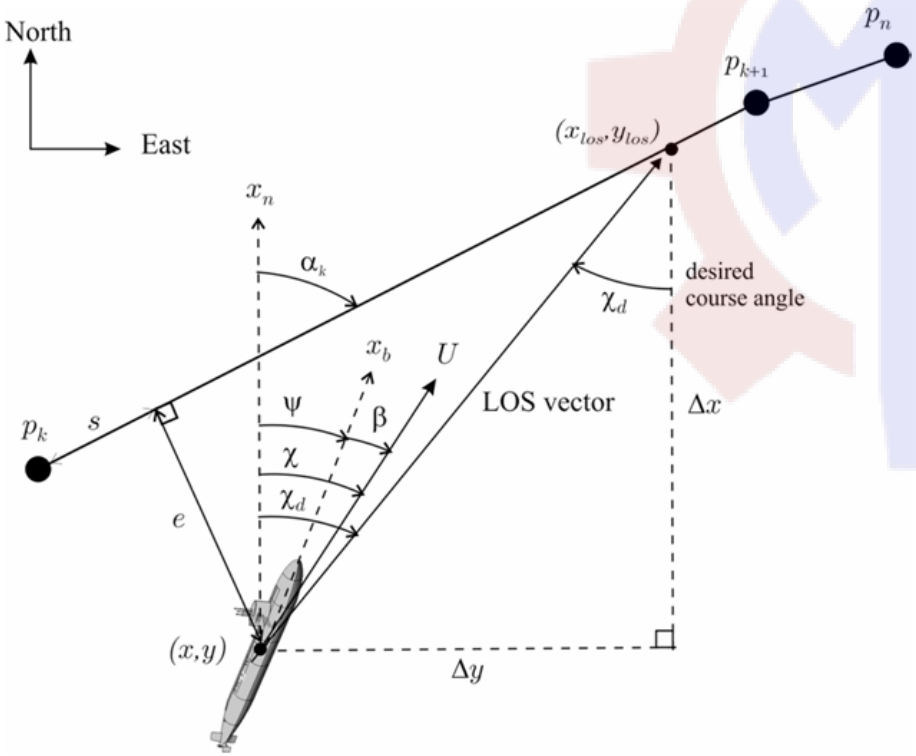
The shortest path (minimum time) between two configurations (x, y, ψ) of a craft moving at constant speed U is a path formed by straight lines and circular arc segments.

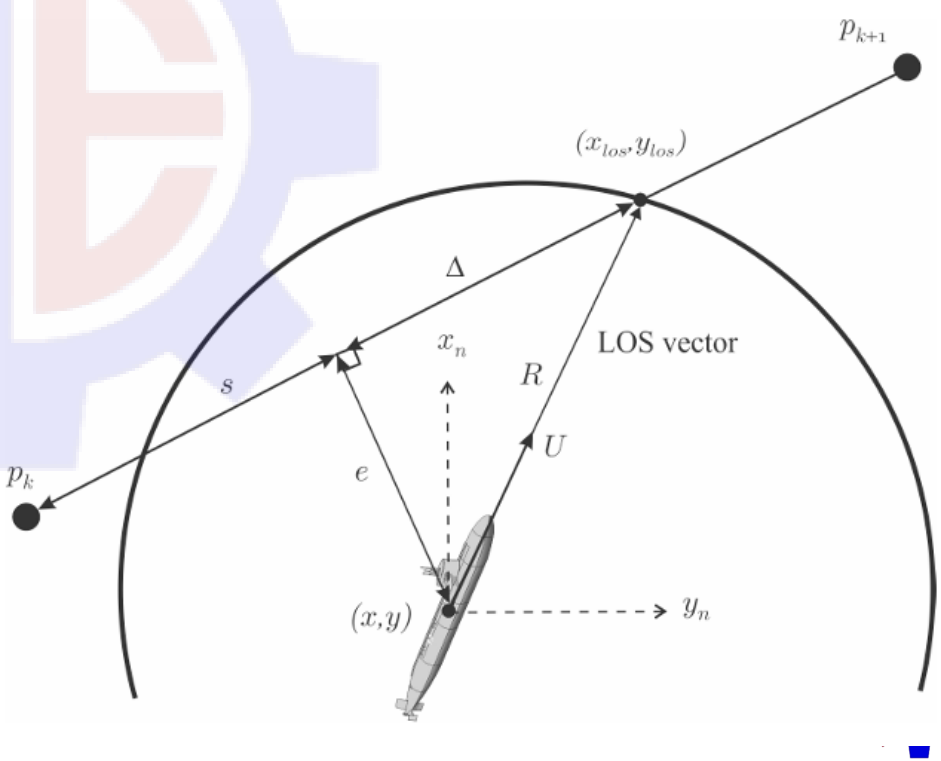


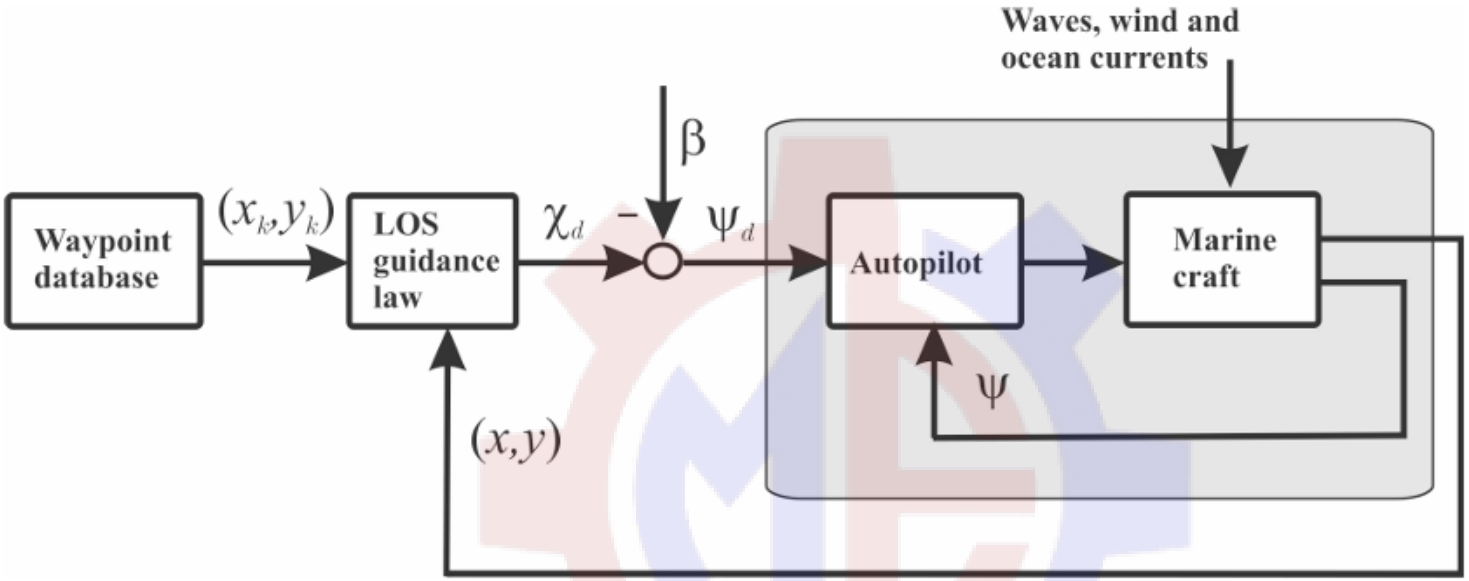


First of all, we consider **a Straight-line path** and introduce two different guidance principles, which could be used to steer along the LOS vector:

- Enclosure-based steering $\rightarrow$ find (x_{los}, y_{los}) base on a a circle with radius $R > 0$ enclosuring (x, y)
- Lookahead-based steering $\rightarrow$ steering law could be easy to modify





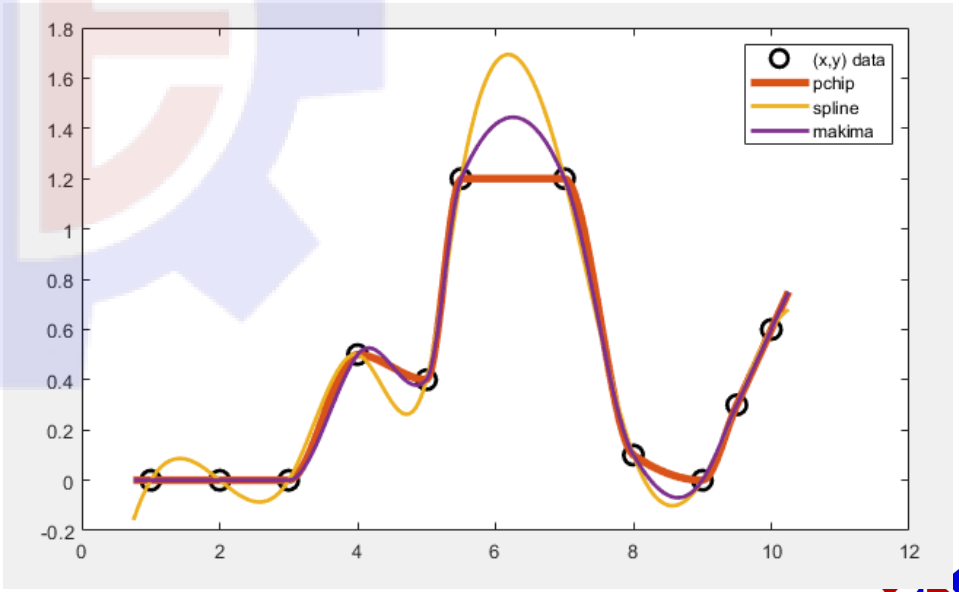


- Enclosure-based steering law $\rightarrow \psi_d = atan2(y_{los} - y, x_{los} - x)$
- Lookahead-based steering law $\rightarrow \psi_d = \chi_d - \beta = \alpha_k + arctan(-\text{PID}(e)) - \beta$

Repeat again: β is the sideslip angle, which calculated by $\beta = arcsin(v/U)$

In the following, **Path-Following for Curved Paths** is introduced.
MATLAB provide different methods for interpolation. For example, three methods for path generation are the cubic spline interpolant (`spline.m`), piecewise cubic Hermite interpolanting polynomial (`pchip.m`) and modified Akima piecewise cubic Hermite interpolation (`makima.m`)

```
x = [1 2 3 4 5 5.5 7 8 9 9.5 10];
y = [0 0 0 0.5 0.4 1.2 1.2 0.1 0 0.3 0.6];
xq = 0.75:0.05:10.25;
yqs = spline(x,y,xq);
yqp = pchip(x,y,xq);
yqm = makima(x,y,xq);

plot(x,y,'ko','LineWidth',2,'MarkerSize',10)
hold on
plot(xq,yqp,'LineWidth',4)
plot(xq,yqs,xq,yqm,'LineWidth',2)
legend('(x,y) data','pchip','spline','makima')
```



In other approach, which is called cubic polynomials, we consider a parametrized path as follows

$$x_d(\varpi) = \sum_{i=0}^{3} a_i \varpi^i; \qquad y_d(\varpi) = \sum_{i=0}^{3} b_i \varpi^i \tag{98}$$

Then, the above constraints must be satisfied

$$x_d(\varpi_{k-1}) = x_{k-1}, \quad x_d(\varpi_k) = x_k, \tag{99}$$

$$\lim_{\varpi \to \varpi_k^-} \frac{d\, x_d(\varpi_k)}{d\, \varpi_k} = \lim_{\varpi \to \varpi_k^+} \frac{d\, x_d(\varpi_k)}{d\, \varpi_k} \tag{100}$$

$$\lim_{\varpi \to \varpi_k^-} \frac{d^2\, x_d(\varpi_k)}{(d\, \varpi_k)^2} = \lim_{\varpi \to \varpi_k^+} \frac{d^2\, x_d(\varpi_k)}{(d\, \varpi_k)^2} \tag{101}$$

$$\frac{d\, x_d(\varpi_k)}{d\, \varpi_k}(\varpi_0) = const, \quad \frac{d\, x_d(\varpi_k)}{d\, \varpi_k}(\varpi_n) = const \tag{102}$$

$$OR \quad \left(\frac{d^2\, x_d(\varpi_k)}{(d\, \varpi_k)^2}(\varpi_0) = const, \quad \frac{d^2\, x_d(\varpi_k)}{(d\, \varpi_k)^2}(\varpi_n) = const \right) \tag{103}$$

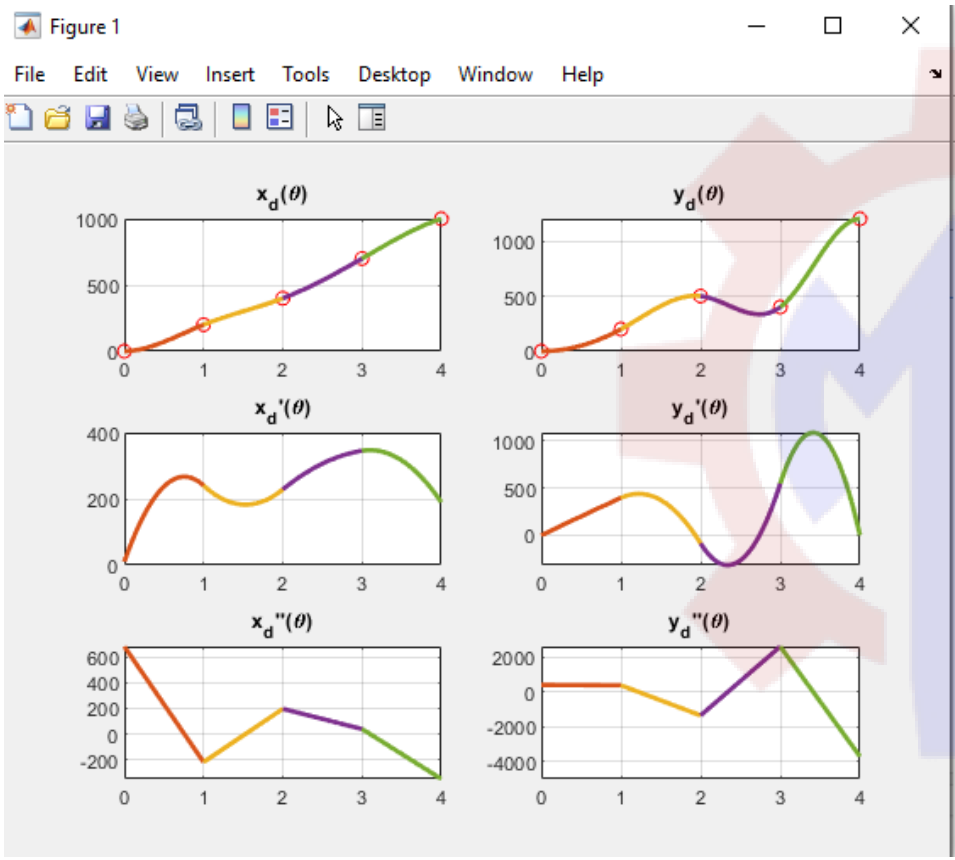


Figure 1
File Edit View Insert Tools Desktop Window Help
$x_d(\theta)$
$y_d(\theta)$
$x_d'(\theta)$
$y_d'(\theta)$
$x_d''(\theta)$
$y_d''(\theta)$

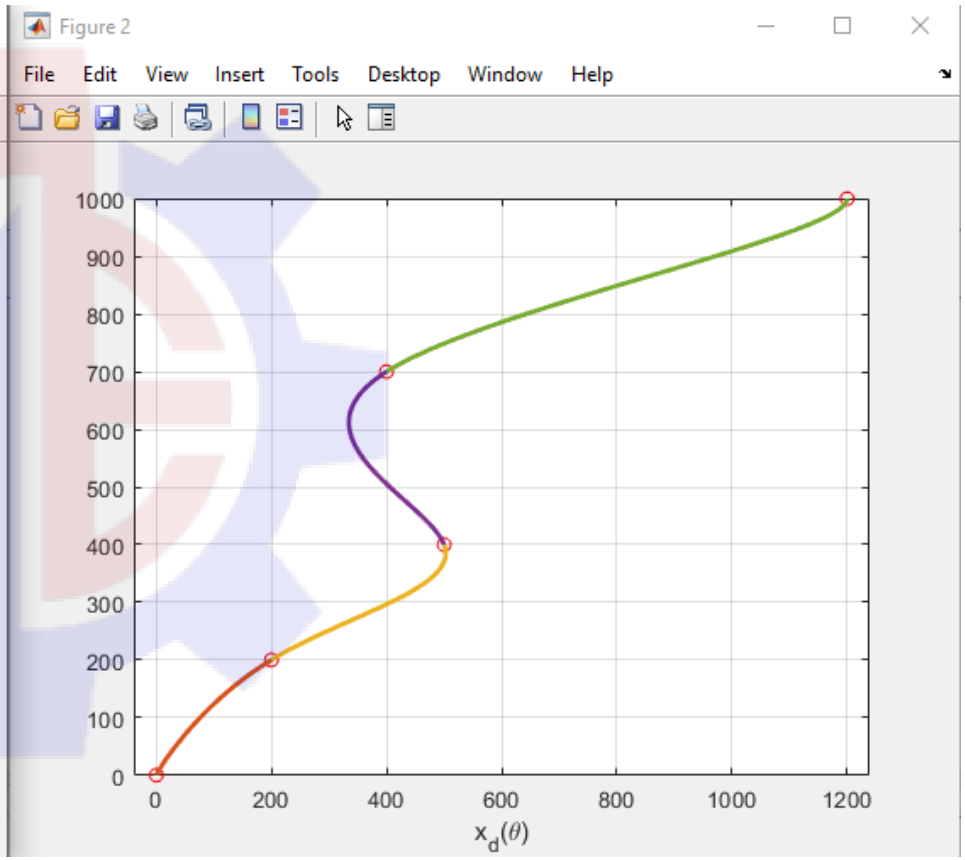


Figure 2
File Edit View Insert Tools Desktop Window Help
$x_d(\theta)$

Transformation of Path to Reference Trajectories using Desired Speed Profiles

$$\dot{\varpi}(t) = \frac{U_d(t)}{\sqrt{\frac{d\,x_d(\varpi_k)}{d\,\varpi_k}(\varpi)^2 + \frac{d\,x_d(\varpi_k)}{d\,\varpi_k}(\varpi)^2}} \tag{104}$$

$$T\dot{U}_d(t) + U_d(t) = U_{ref} \tag{105}$$

In another view, the cubic polynomials approach could be formed by the **Nonlinear Constrained Optimization Problem**. Firstly, the constraint of the cubic polynomials approach could be represented by the the linear form

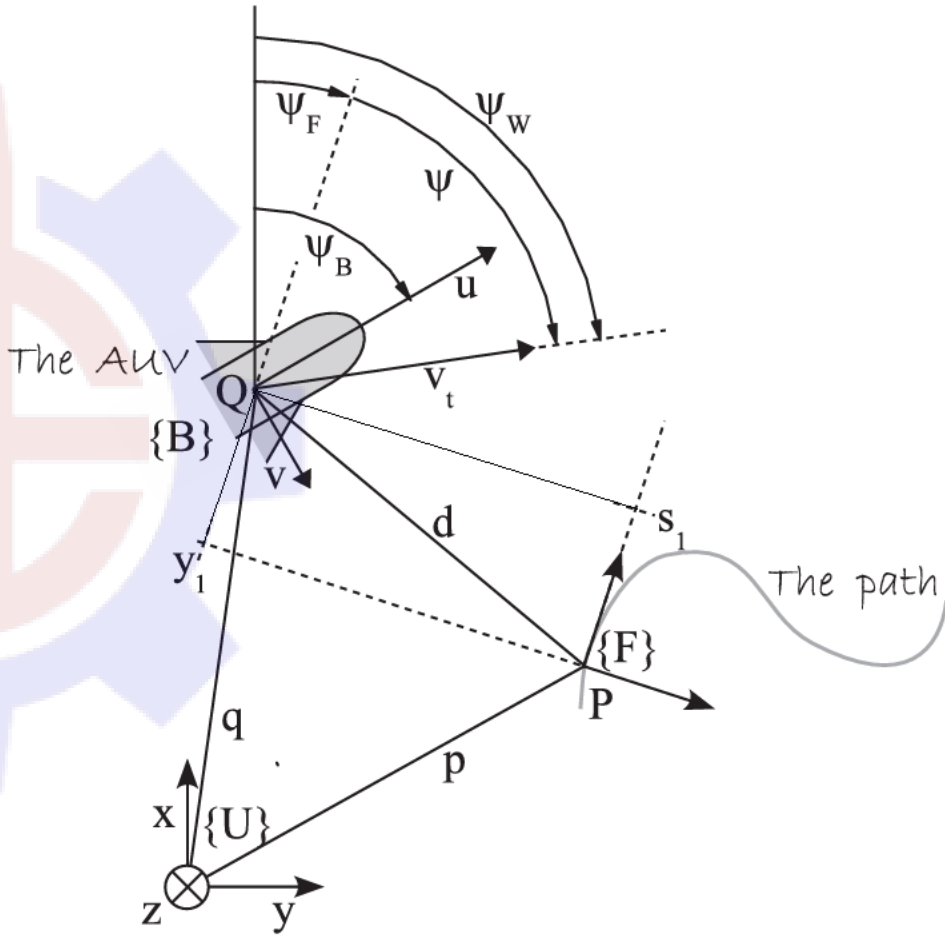
$$y = A(t_k, t_{k+1})x \tag{106}$$

where x is the vector of polynomial coefficients, and y, A are the derived constraints vector and matrix. Then, the optimization problem is formed

$$\text{minimize} \Big[\Big(A(t_k, t_{k+1})x - y \Big)^\top \Big(A(t_k, t_{k+1})x - y \Big) \Big] \tag{107}$$

with the constraint of velocities and accelerations.

The solution to the problem of path-following proposed here builds on the following intuitive explanation a simple path-following controller should compute (i) the distance between the vehicle's center of mass Q and a point P, and (ii) the angle between the vehicle's total velocity vector v_t and the tangent to the path at P, and reduce both to zero. This motivates the development of the 'kinematic' model of the vehicle in terms of a Serret–Frenet frame $\{F\}$ that moves along the path; $\{F\}$ plays the role of the body axis of a 'virtual target vehicle' that should be tracked by the 'real vehicle'.



$$\dot{s}_1 = -\dot{s}(1 - c_c y_1) + v_t \cos\psi, \quad (108)$$

$$\dot{y}_1 = -c_c \dot{s}_1 + v_t \sin\psi, \quad (109)$$

$$\dot{\psi} = r + \dot{\beta} - c_c \dot{s}. \quad (110)$$

Assuming we have a precise estimation of the function $\mu(s)$, and given s, we compute

$$x_s(\mu) = \sum_{i=1}^{n} a_i \mu^i, \quad y_s(\mu) = \sum_{i=1}^{n} b_i \mu^i, \quad (111)$$

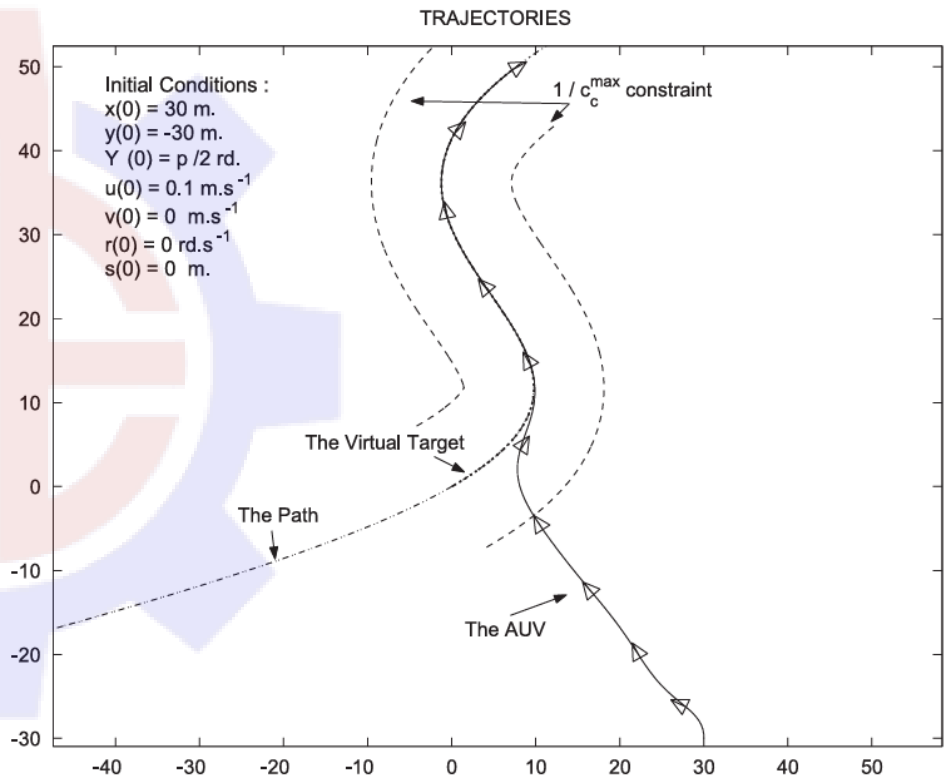
$$\psi_F(s) = \arctan\left(\frac{(y_s)'}{(x_s)'}\right), \quad (112)$$

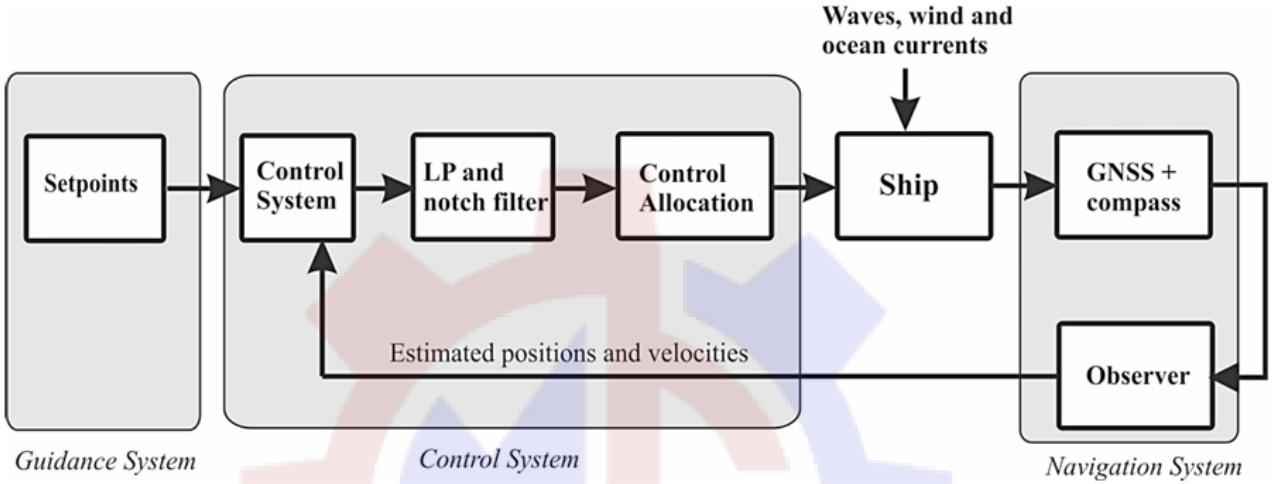
$$c_c(s) = \frac{\partial \psi_F(s)}{\partial \mu}\frac{d\mu}{ds}, \quad (113)$$

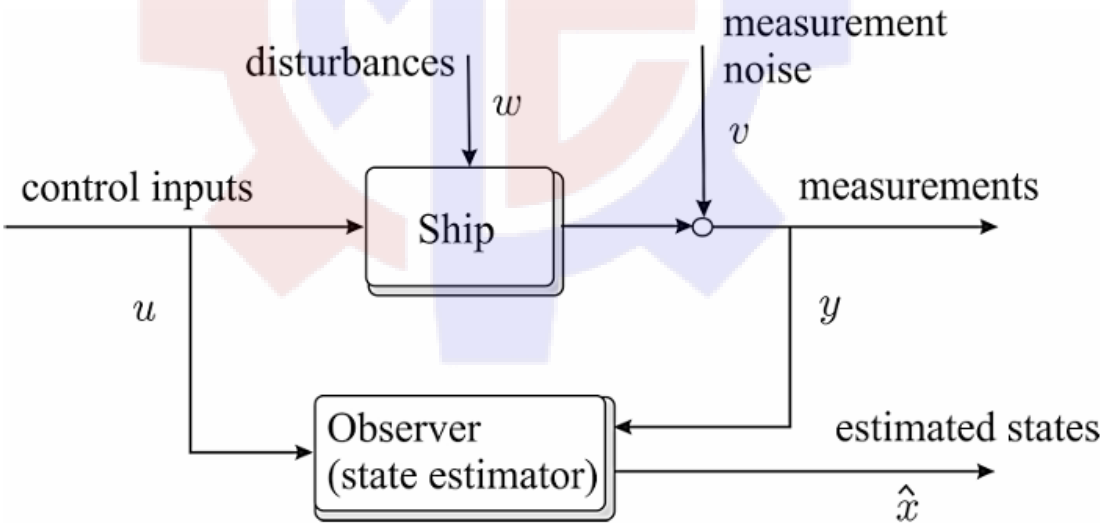
$$\frac{\partial c_c(s)}{\partial s} = \frac{\partial c_c(s)}{\partial \mu}\frac{d\mu}{ds}, \quad (114)$$

$$(x_s)' = \frac{dx_s}{d\mu}, \quad (y_s)' = \frac{dy_s}{d\mu} \quad (115)$$

$$\frac{d\mu}{ds} = \frac{1}{\sqrt{[(x_s)']^2 + [(y_s)']^2}}. \quad (116)$$




Waves, wind and ocean currents
Setpoints
Control System
LP and notch filter
Control Allocation
Ship
GNSS + compass
Estimated positions and velocities
Observer
Guidance System
Control System
Navigation System


disturbances
w
measurement noise
v
control inputs
Ship
measurements
u
y
Observer (state estimator)
estimated states
$\hat{x}$

Low-Pass filtering: $\omega_b \ll \omega_e$,

$$\hat{y}(s) = h_{lp}(s)y(s) \tag{117}$$

where $h_{lp}(s)$ could be chosen as first-order low-pass filter, $h_{lp}(s) = \dfrac{1}{1+T_f s}$ or the higher-order low-pass filter by using Butterworth filter, $h_{lf}(s) = \dfrac{1}{p(s)}$, with $p(s)p(-s) = 1 + (s/j\omega_f)^{2n}$

Cascade Low-Pass and Notch filtering:

$$\hat{y}(s) = h_{lp}(s)h_n(s)y(s) \tag{118}$$

with

$$h_n(s) = \frac{s^2 + 2\xi\omega_n s + \omega_n^2}{(s+\omega_n)^2}; \quad OR \quad h_n(s) = \prod_{i=1}^{3} \frac{s^2 + 2\xi\omega_i s + \omega_i^2}{(s+\omega_i)^2} \tag{119}$$

Consider the linear time-invariant (LTI) system,

$$\dot{x} = Ax + Bu \tag{120}$$

$$y = Cx \tag{121}$$

and consider its dual system

$$\dot{\tilde{x}} = A^\top \tilde{x} + C^\top \tilde{u} \tag{122}$$

$$\tilde{y} = B^\top \tilde{x} \tag{123}$$

Luenberger observer: coud be designed as **pole placement control design** for the dual system.

Kalman observer: coud be designed as **LQR control design** for the dual system.

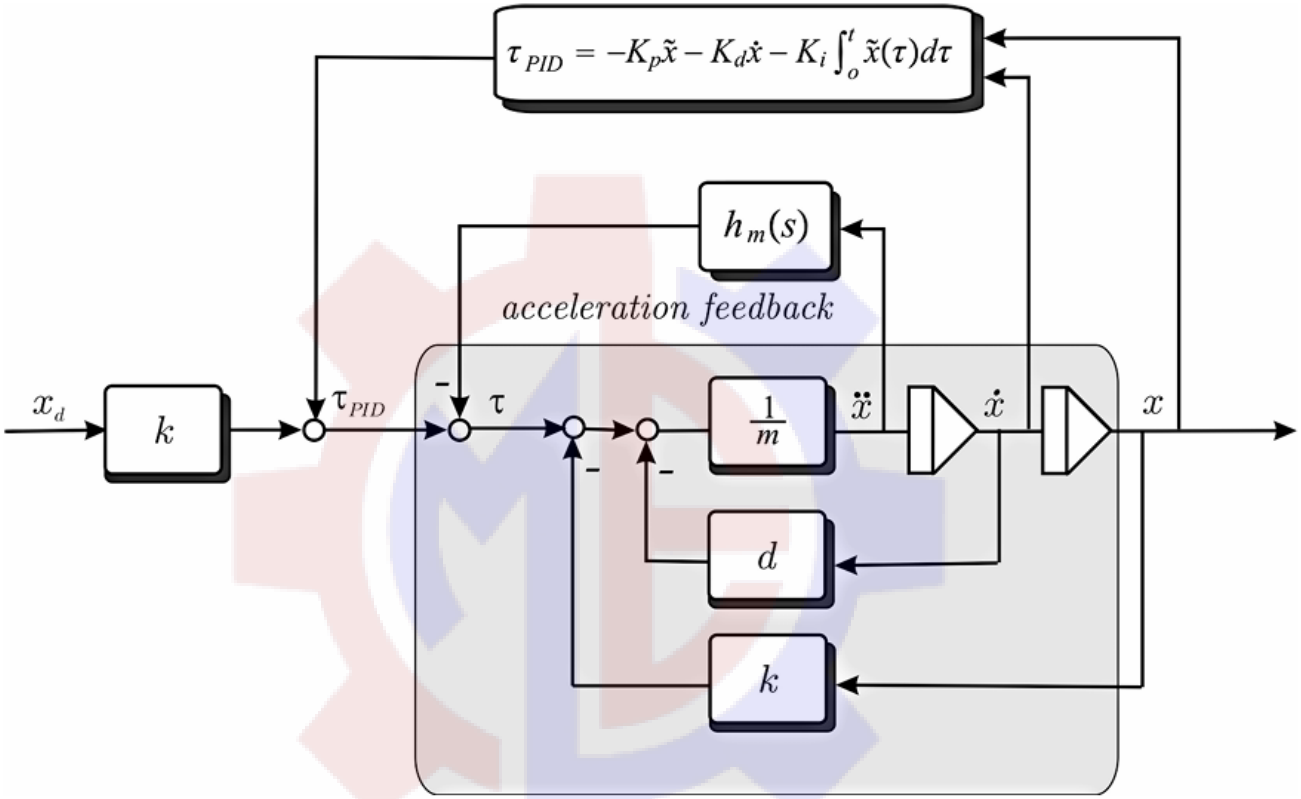
Both of above-mentioned observers follow the bellow structure

$$\dot{z} = Az + Bu + L(y - Cz) \tag{124}$$

with L is observer gain.

In other way, the advance observers that we could approach

- Extended Kalman Filter
- High-gain observer
- Sliding Mode Observer
- Extended State observer
- Fixed- and Finite-time observer
- ...



Closed-loop feedback system - PID control approach

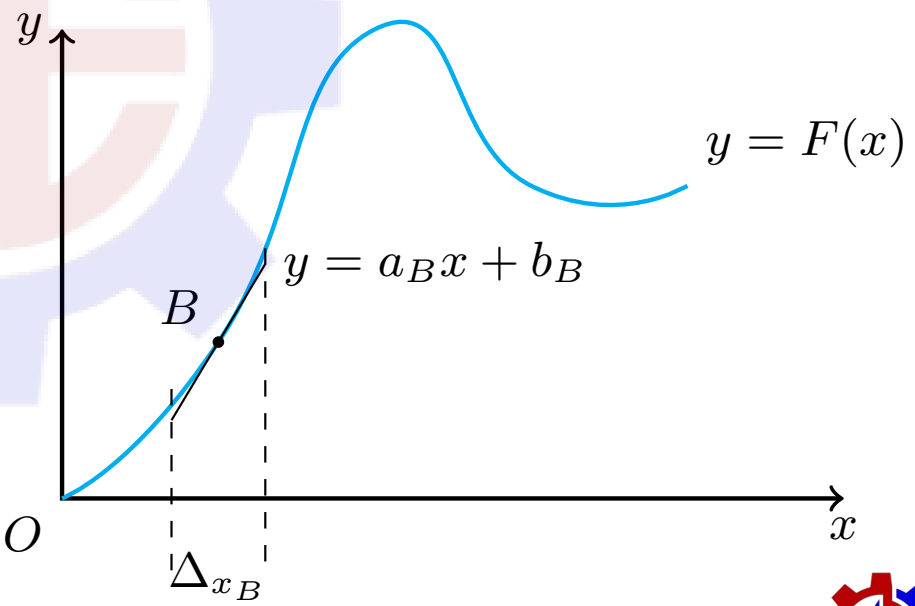
$$\tau_{PID} = -K_p e - K_d \dot{e} - K_i \int_0^t e(\tau)d\tau \tag{125}$$

In the below section, we approach the control problem of linear system by Linear control technique.
First of all, consider the linearization algorithm for nonlinear system,

Assume that, the linear function at B is

$$y = a_B x + b_B \tag{126}$$

$$\approx F(x)\Big|_{x \in \Delta_{x_B}} \tag{127}$$



where a_B is the coefficient of first degree, easy to determine

$$a_B = \frac{\partial F(x)}{\partial x}\Big|_{x=x_B} \tag{128}$$

and b_B is the bias.

Consider a general system, which found in the above

$$\dot{x} = f(x) + g(x)\tau = F(x, \tau) \tag{129}$$

$\rightarrow$ Let's approximate it to the linear original difference equation (ODE)

$$\dot{x} = Ax + B\tau \tag{130}$$

where, the state-space matrices are defined as

$$A = \left.\frac{\partial F(x,\tau)}{\partial x}\right|_{x=x_e,\tau=\tau_e} \tag{131}$$

$$B = \left.\frac{\partial F(x,\tau)}{\partial \tau}\right|_{x=x_e,\tau=\tau_e} \tag{132}$$

In the following step, we propose a linear control technique - pole placement control.

First, consider the state-space model, described as follows

$$\dot{x} = Ax + B\tau \tag{133}$$

As mention above, the control signal is

$$\tau = h(x) \tag{134}$$

such as, the system in (133) is stable.
For simplify the control signal, we assume that, the control law is a gain of state variables

$$\tau = h(x) = -Kx \tag{135}$$

This is the most simplify of control law!

With the control signal defined as $\tau = h(x) = -Kx$, the close-loop system becomes

$$\dot{x} = Ax - BKx \tag{136}$$

$$= (A - BK)x \tag{137}$$

Let $\tilde{A} = A - BK$. This ODE exist the solution as

$$x = \mathrm{e}^{\tilde{A}t} = \sum_{i=0}^{\infty} \frac{\tilde{A}^i t^i}{i!} = P^{-1}\Big(\sum_{i=0}^{\infty} \frac{L^i t^i}{i!}\Big)P = P^{-1}\mathrm{e}^{Lt}P \tag{138}$$

where

$$L = diag(\lambda_1, \lambda_2, ..., \lambda_i) \tag{139}$$

λ_i are the eigenvalue of $\tilde{A}$ with eigenvector P_i and

$$P = \begin{bmatrix} P_1^\top & P_2^\top & ... & P_i^\top \end{bmatrix}^\top \tag{140}$$

Find K such as $\lim_{t\to\infty} x = 0$ that mean Find K such as $L < 0$

Similarly, by considering a optimization problem, Linear Quadratic Regulator is constructed.

First of all, consider linear time-invariant system

$$\frac{d}{dt}x = Ax + Bu \tag{141}$$

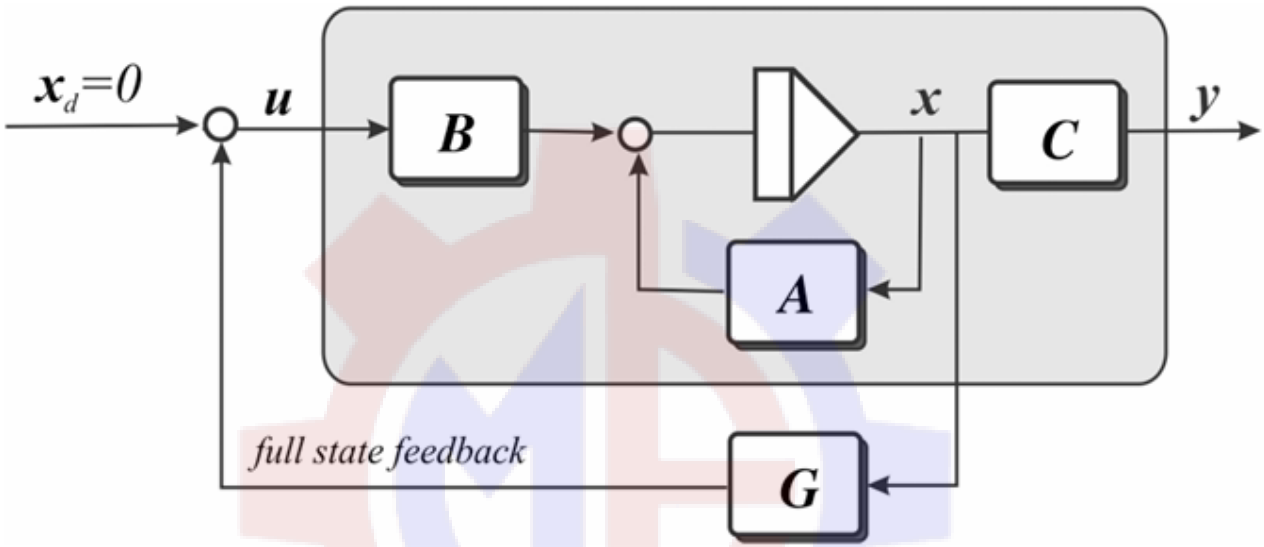
and control signal for this system

$$u = \omega - Gx \tag{142}$$

The LQR controller is a state feedback controller, with the control signal u found to be satisfied

$$J = \frac{1}{2}\int_0^\infty \left(x^T Q x + u^T R u\right)\mathrm{d}t \to \min \tag{143}$$

J is the cost function, $Q = Q^T > 0$ and $R = R^T > 0$ are weight matrices.



The first step to find the LQR controller is to find the solution $P = P^T > 0$ of the Algebraic Riccati Equation (ARE)

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{144}$$

then, the controller matrix G can be found

$$G = R^{-1}B^T P \tag{145}$$

In the following section, a basic nonlinear control strategy that based on Sliding Mode Control is introduced.

For nonlinear control strategy, we consider the Sliding Mode Control technique. First of all, the control purpose is that the state variables track the desired states. That means if we consider the system, which is described by the above ODE

$$\ddot{x} = f(x, \dot{x}) + g(x, \dot{x})\tau \tag{146}$$

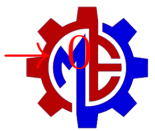
Assume that, the desired states is defined by x_d. Then, the error states are

$$e = x - x_d; \quad \dot{e} = \dot{x} - \dot{x}_d \tag{147}$$

Then, we define a sliding variable and a sliding surface with positive definite matrix $k > 0$ as follows

$$s = \dot{e} + ke; \quad \mathcal{S} = \{x, \dot{x} | s = 0\} \tag{148}$$

Notice that, by choosing a positive definite matrix k, this is easy to achieve that $s \to 0$ then $e \to 0$ and $\dot{e} \to 0$



In the following step, a control law is proposed to act the sliding variables to the sliding surface. First, we consider the Lyapunov candidate as follows

$$V = \frac{1}{2} s^\top s \quad > 0 \tag{149}$$

And its derivative could be easy to achieve

$$\begin{aligned} \dot{V} &= s^\top \dot{s} = s^\top (\ddot{x} - \ddot{x}_d + k\dot{e}) \\ &= s^\top (f(x, \dot{x}) + g(x, \dot{x})\tau - \ddot{x}_d + k\dot{e}) \end{aligned} \tag{150}$$

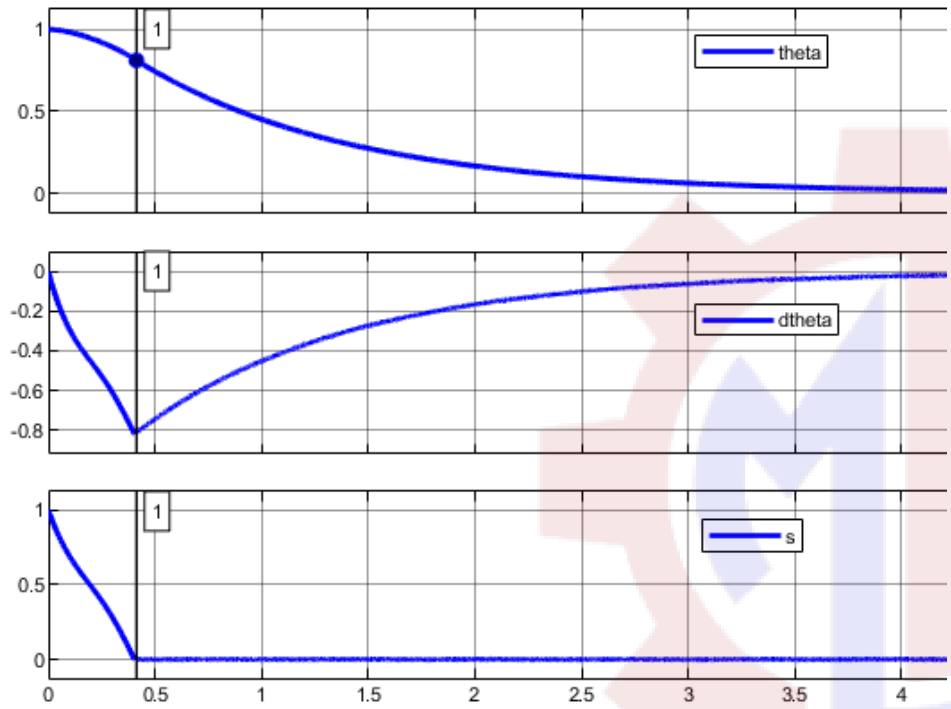
Following Lyapunov stability theorem, let's choose the control signal as

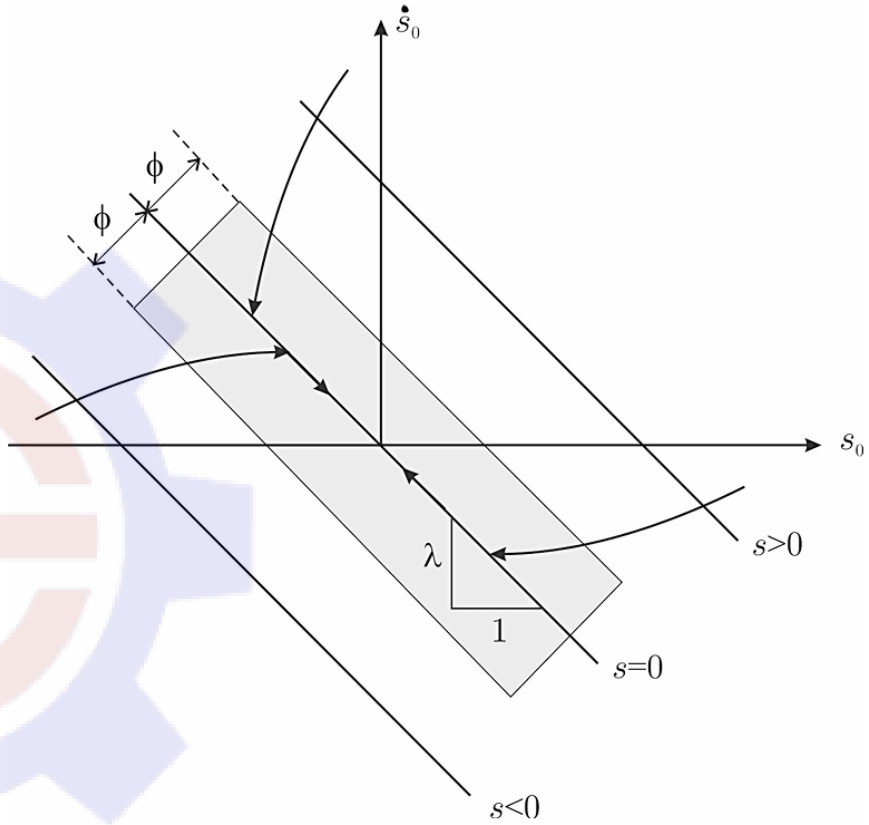
$$\tau = g(x, \dot{x})^{-1} \big(- f(x, \dot{x}) + \ddot{x}_d - k\dot{e} - \eta s - \gamma sign(s) \big) \tag{151}$$

then, the derivative of Lyapunov candidate becomes

$$\dot{V} = s^\top (-\eta s - \gamma sign(s)) = -s^\top \eta s - \gamma |s| \tag{152}$$

$\rightarrow$ if η and γ are positive definite, $\dot{V} < 0$, thus, $s \rightarrow 0$





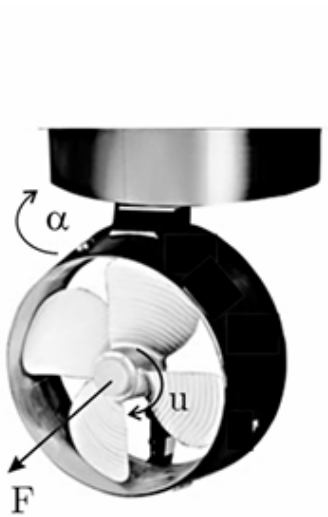
- Phase 1: The sliding variables reach the sliding surface, where the sliding variables are zeros
- Phase 2: The state errors reach to zeros such that $\dot{e} + ke = 0$

Need a map from actuator to Force and Moment. That means

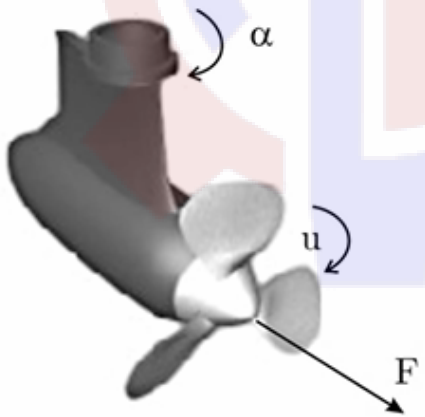
$$\text{Forces and Moments at center of mass: } \tau \in \mathbb{R}^6 \tag{153}$$

$$\text{Actuators: } u \in \mathbb{R}^n \tag{154}$$

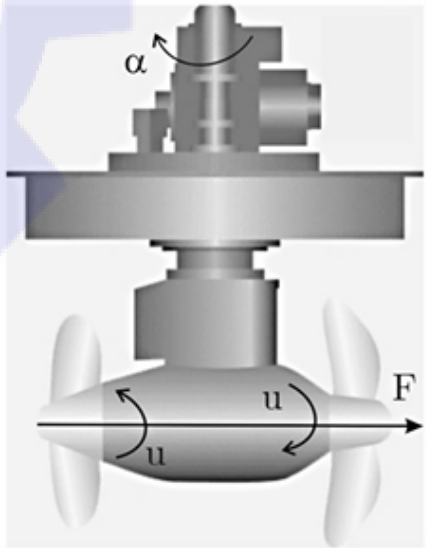
$$\text{Forces and Moments allocation: } f : \mathbb{R}^n \mapsto \mathbb{R}^6 \tag{155}$$



Azimuth thruster



Podded propeller



Contra-rotating propeller

Forces and Moment constraint: $F_{\vec{k}} = \sum_i F_i \vec{k}$ (156)

$$M_{\vec{k}} = \sum_i M_i \vec{k} \quad (157)$$

$$M \text{ [Nm]} = F \text{ [N]} \times l \text{ [m]} \quad (158)$$

u_1, α_1	*fore azimuth thruster*	u_4	*aft tunnel thruster*
u_2	*fore tunnel thruster*	u_5	*starboard main propeller*
u_3, α_2	*aft azimuth thruster*	u_6	*port main propeller*

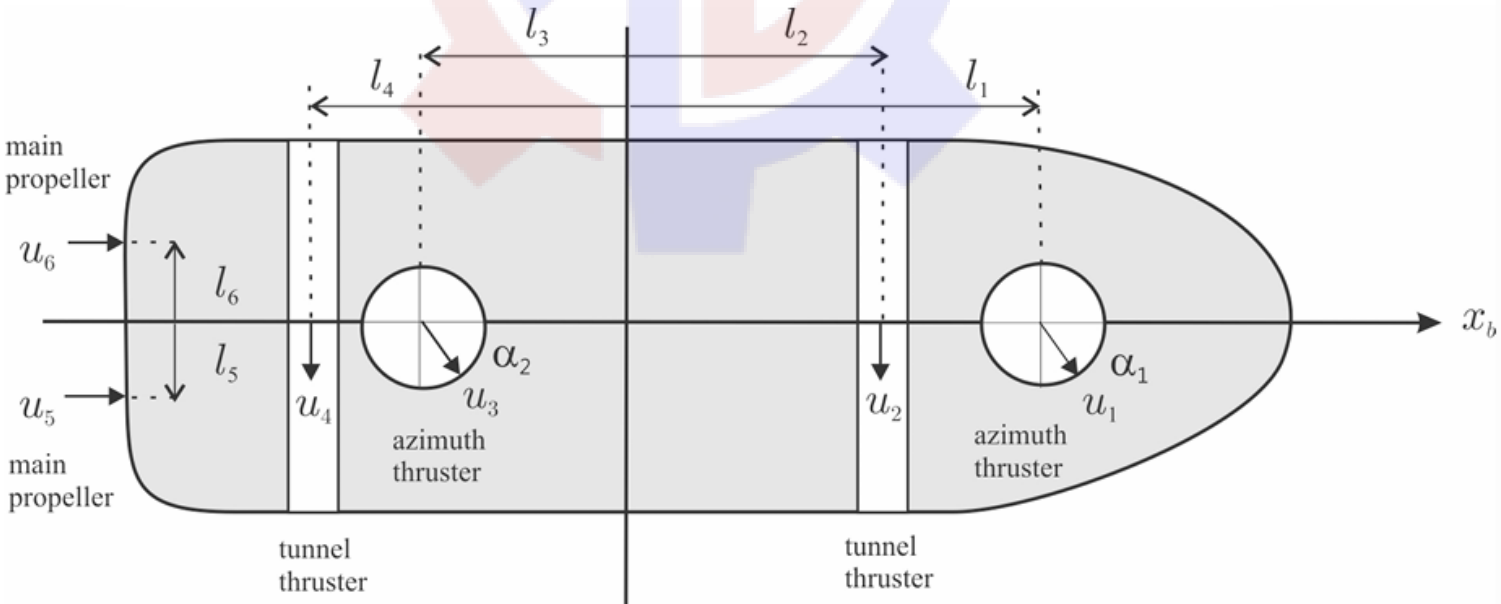
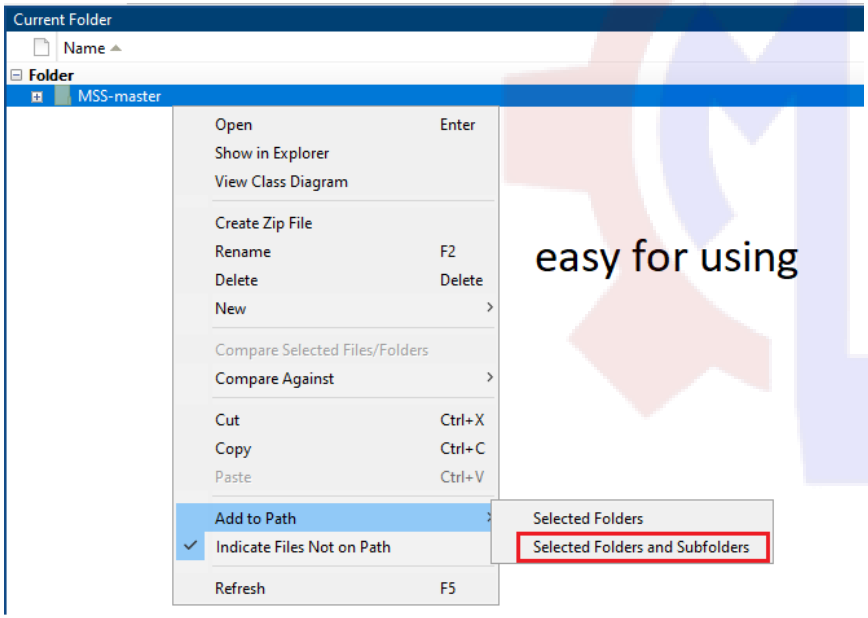
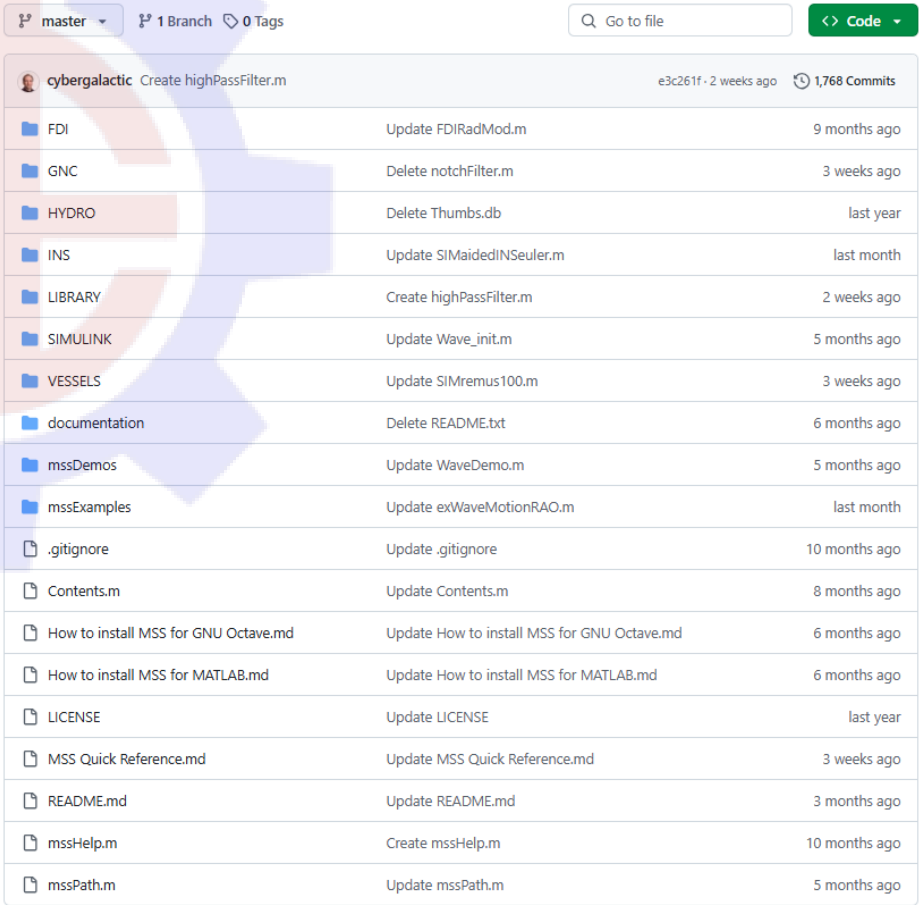


Table of Contents

The Marine Systems Simulator (MSS) is a Matlab and Simulink library for marine control systems design. The m-files are compatible with the free software GNU Octave. MSS includes models for ships, underwater vehicles, uncrewed surface vehicles, and floating structures. The library also contains guidance, navigation, and control (GNC) blocks for time-domain simulation.



easy for using



Numerical simulations

MSS Toolbox

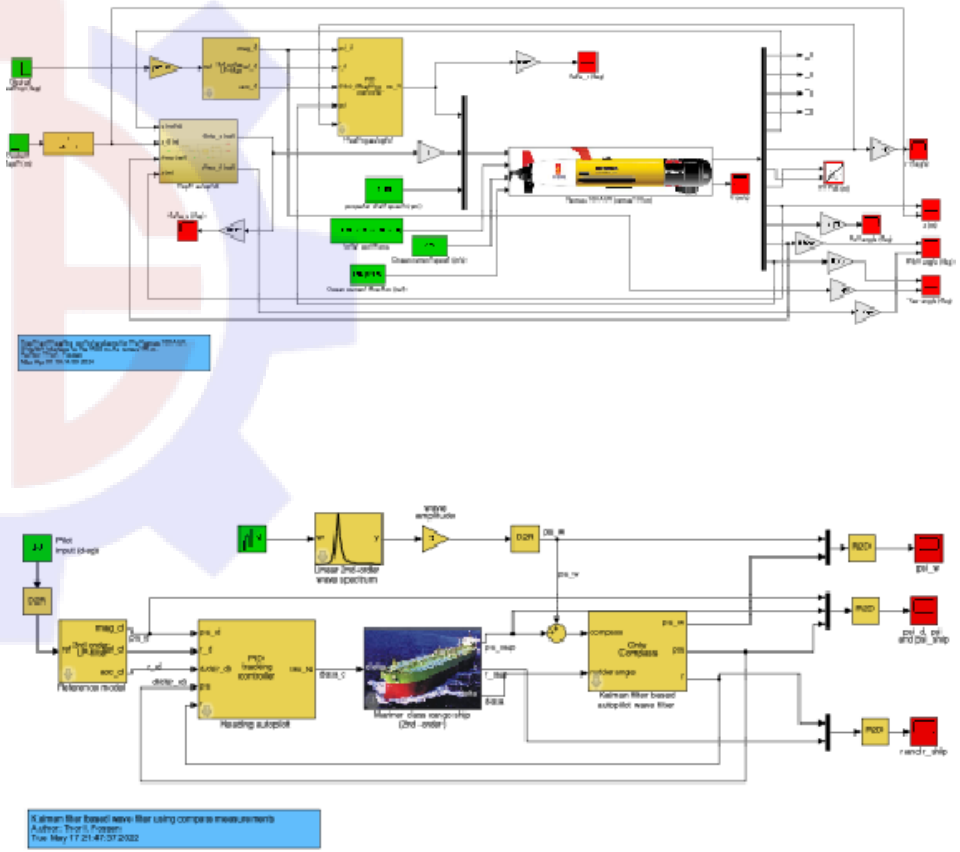
The Marine Systems Simulator (MSS) is a Matlab and Simulink library for marine control systems design. The m-files are compatible with the free software GNU Octave. MSS includes models for ships, underwater vehicles, uncrewed surface vehicles, and floating structures. The library also contains guidance, navigation, and control (GNC) blocks for time-domain simulation.

```
% Script for time-series simulations:
%
% SIMclarke83     - Simulate clarke83.m under PD control.
% SIMdsrv         - Simulate DSRV.m with an autopilot for depth control
%                   using successive-loop closure.
% SIMfrigate      - Simulate frigate.m using a PID heading autopilot.
% SIMmariner      - Simulate mariner.m under PD control.
% SIMotter        - Simulate otter.m under feedback control.
% SIMosv          - Simulate osv.m under nonlinear DP control with
%                   constrained control allocation (dynamic optimization).
% SIMcontainer    - Simulate container.m and Lcontainer.m under PD control.
% SIMnavalvessel  - Simulate navalvessel.m under PD control.
% SIMnpsauv       - Simulate npsauv.m with MIMO PID autopilots for depth
%                   and heading control, and 3-D straight-line path
%                   following using ALOS
% SIMremus100     - Simulate remus100.m using autopilots for depth and
%                   heading control, and adaptive line-of-sight (ALOS)
%                   guidance laws for 3-D path-following control.
% SIMrig          - Simulate the 6-DOF semisubmersible model under PID
%                   control.
% SIMsupply       - Simulate the linear supply vessel model under DP control.
% SIMtanker       - Simulate the tanker model under DP control.
```

```
% The MSS marine craft models are compatibel with MATLAB and the free
% software GNU Octave (www.octave.org):
%
% clarke83        - Linear maneuvering model parametrized using (L,B,T)
%                   found from linear regression of model tests (Clarke et
%                   al. 1983).
% container       - Nonlinear maneuvering model of a high-speed container
%                   ship, L = 175 m, including roll (Son and Nomoto 1982)
% DSRV            - Deep submergence rescue vehicle (DSRV), L = 5.0 m
%                   (Healey 1992).
% frigate         - Nonlinear autopilot model for a frigate, L = 100 m.
% Lcontainer      - Linearized model of a high-speed container ship,
%                   L = 175 m, including the roll mode (Son and Nomoto 1982).
% mariner         - Nonlinear maneuvering model for the Mariner class
%                   vessel, L = 160 m.
% navalvessel     - Nonlinear maneuvering model of a multipurpose naval
%                   vessel, L = 51.5 m.
% npsauv          - Naval Postgraduate School autonomous underwater vehicle
%                   (AUV), L = 5.3 m.
% osv             - Nonlinear model of an Offshore supply vessel (OSV),
%                   L = 83.0 m.
% otter           - OTTER small autonomous USV, L = 2.0 m.
% remus100        - REMUS 100 autonomous underwater vehicle (AUV), L = 1.9 m
% rig             - Semisubmersible linear mass-damper-spring model,
%                   L = 84.6 m.
% supply          - Linear DP model of a supply vessel, L = 76.2 m.
% tanker          - Nonlinear course unstable maneuvering model of a
%                   tanker, L = 304 m.
% zeefakkel       - Nonlinear autopilot model of a pleasure craft, L = 45 m.
```

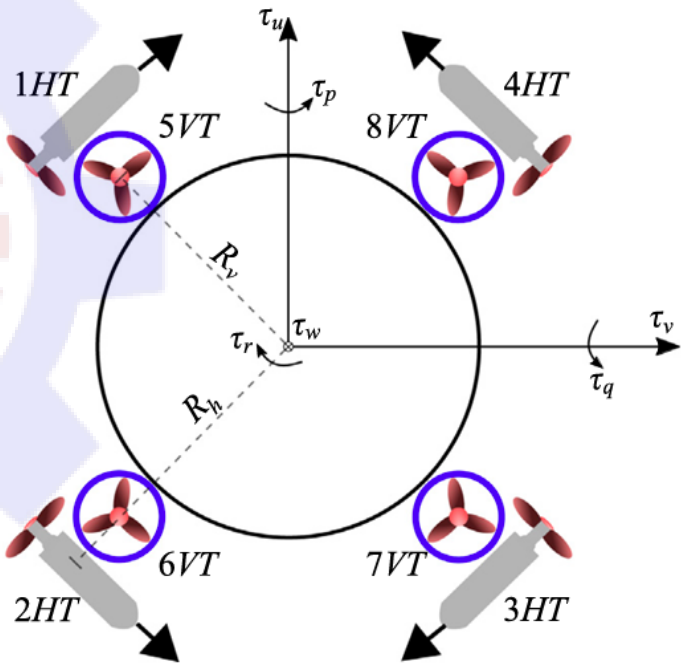
Simulink models are constructed by MATLAB R2024

- SIMULINK
  - figs
  - init
  - mssSimulinkDemos
    - demoAUVdepthHeadingControl.slx
    - demoCS2passiveObserverDP.slx
    - demoDPThrusterModels.slx
    - demoDSRVdepthControl.slx
    - demoKalmanWavefilterAutop.slx
    - demoMarinerPathFollowingCourseControl.slx
    - demoNavalVesselMano.slx
    - demoNPSAUV.slx
    - demoOtterUSVHeadingControl.slx
    - demoOtterUSVPathFollowingCourseControl.slx
    - demoOtterUSVPathFollowingHeadingControl.slx
    - demoPanamaxContainerShip.slx
    - demoPassiveWavefilterAutopilot1.slx
    - demoPassiveWavefilterAutopilot2.slx
    - demoS175WindCurrentAutopilot.slx
    - demoSemisubDPsystem.slx
    - demoWaveElevation.slx
    - demoWaypointGuidance.slx
  - mssWamitShipXTemplates
  - mssSimulink.slx

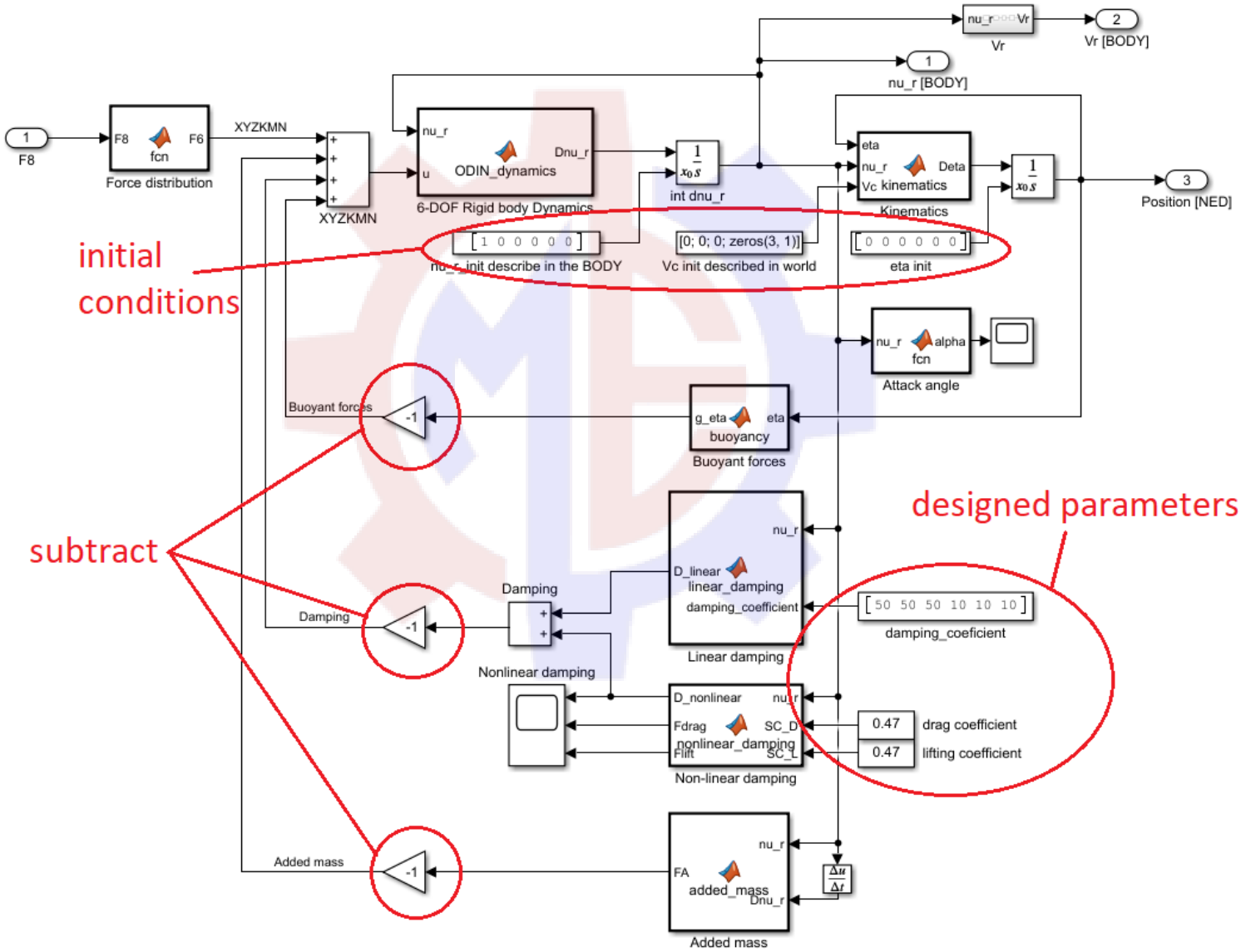


Forces distribution

```
function F6 = fcn(F8)
s = 1/sqrt(2);
Rver = 0.63;
Rhor = 0.89;
E = [s -s -s s 0 0 0 0;
     s s -s -s 0 0 0 0;
     0 0 0 0 -1 -1 -1 -1;
     0 0 0 0 Rver*s Rver*s -Rver*s -Rver*s;
     0 0 0 0 Rver*s -Rver*s -Rver*s Rver*s;
     Rhor -Rhor Rhor -Rhor 0 0 0 0];

F6 = E*F8;
```



Simulink simulation diagram



Dynamics and kinematics m-file of ODIN

```
function Dnu_r = ODIN_dynamics(nu_r, u)
%% Model parameters
rho = 1025;
g = 9.81;
R = 0.5;
V = 4/3*pi*R^3;
B = rho*g*V;
W = B;
m = W/g;
% Moment of inertia
I = (2/5) * m * R^2;
Ib = diag([I, I, I]);
rgb = [0; 0; 0];
%% Model dynamics
I3 = eye(3);
O3 = zeros(3);
nu1 = nu_r(1:3);
nu2 = nu_r(4:6);
skewMatrix = @(x) [0 -x(3) x(2); x(3) 0 -x(1); -x(2) x(1) 0];
%% Described in center of gravity (COM)
MRB_G = [m*I3 -m*skewMatrix(rgb); m*skewMatrix(rgb) Ib];
CRB_G = [m*skewMatrix(nu2), -m*skewMatrix(nu2)*skewMatrix(rgb);
    skewMatrix(nu2)*skewMatrix(rgb), -skewMatrix(Ib*nu2)];
S = skewMatrix(rgb);
H = [eye(3)     S'
     zeros(3,3) eye(3) ];
%% Described in origin
MRB = H'*MRB_G*H;
CRB = H'*CRB_G*H;
%% M*dnu + C*nu + G = u
Dnu_r = MRB^-1*(u - CRB*nu_r);
end
```

neutral buoyant

frame moving

```
function Deta = kinematics(eta, nu_r, Vc)
I3 = eye(3);
O3 = zeros(3);
phi = eta(4); theta = eta(5); psi = eta(6);

% Rotation matrix about x-axis (Roll)
R_x = [1, 0, 0;
    0, cos(phi), -sin(phi);
    0, sin(phi), cos(phi)];
% Rotation matrix about y-axis (Pitch)
R_y = [cos(theta), 0, sin(theta);
    0, 1, 0;
    -sin(theta), 0, cos(theta)];
% Rotation matrix about z-axis (Yaw)
R_z = [cos(psi), -sin(psi), 0;
    sin(psi), cos(psi), 0;
    0, 0, 1];
% Combined rotation matrix (Z-Y-X convention)
R = R_z * R_y * R_x;

nu = nu_r + [R^(-1) O3; O3 I3]*Vc;

T = [1, sin(phi)*tan(theta), cos(phi)*tan(theta);
    0, cos(phi), -sin(phi);
    0, sin(phi)/cos(theta), cos(phi)/cos(theta)];
J = [R O3;
    O3 T];

Deta = J*nu;
```

NED frame

Buoyancy forces

```
function g_eta = buoyancy(eta)
phi = eta(4); theta = eta(5); psi = eta(6);
rho = 1025;
g = 9.81;
R = 0.5;
V = 4/3*pi*R^3;
B = rho*g*V;
W = B;
% m = W/g;
xg = 0;
yg = 0;
zg = 0;
xb = 0;
yb = 0;
zb = 0;
g_eta = [
     (W - B)*sin(theta);
    -(W - B)*cos(theta)*sin(phi);
    -(W - B)*cos(theta)*cos(phi);
    -(yg*W - yb*B)*cos(theta)*cos(phi) + (zg*W - zb*B)*cos(theta)*sin(phi);
     (zg*W - zb*B)*sin(theta) + (xg*W - xb*B)*cos(theta)*cos(phi);
    -(xg*W - xb*B)*cos(theta)*sin(phi) - (yg*W - yb*B)*sin(theta)];
end
```

position of CO and CB

Damping terms

```
function D_linear = linear_damping(nu_r, damping_coefficient)
Xu = damping_coefficient(1);
Yw = damping_coefficient(2);
Zw = damping_coefficient(3);
Kp = damping_coefficient(4);
Mq = damping_coefficient(5);
Nr = damping_coefficient(6);

D = diag([Xu, Yw, Zw, Kp, Mq, Nr]);
D_linear = D*nu_r;
end
```

design parameters

```
function [D_nonlinear, Fdrag, Flift] = nonlinear_damping(nu_r, SC_D, SC_L)
Vr = sqrt(nu_r(1:3)'*nu_r(1:3));

if nu_r(3) == 0 && nu_r(1) == 0
    alpha = 0;
else
    alpha = atan(nu_r(3)/nu_r(1));
end
rho = 1025;
% SC_D = 0.47;
% SC_L = 0.47;
Fdrag = 1/2 * Vr^2 * rho * SC_D;
Flift = 1/2 * Vr^2 * rho * SC_L;

D_nonlinear = [
    -Flift * sin(alpha) + Fdrag * cos(alpha);
    0;
    Flift * cos(alpha) + Fdrag * sin(alpha);
    zeros(3, 1)];  % 3x1 zero vector
end
```

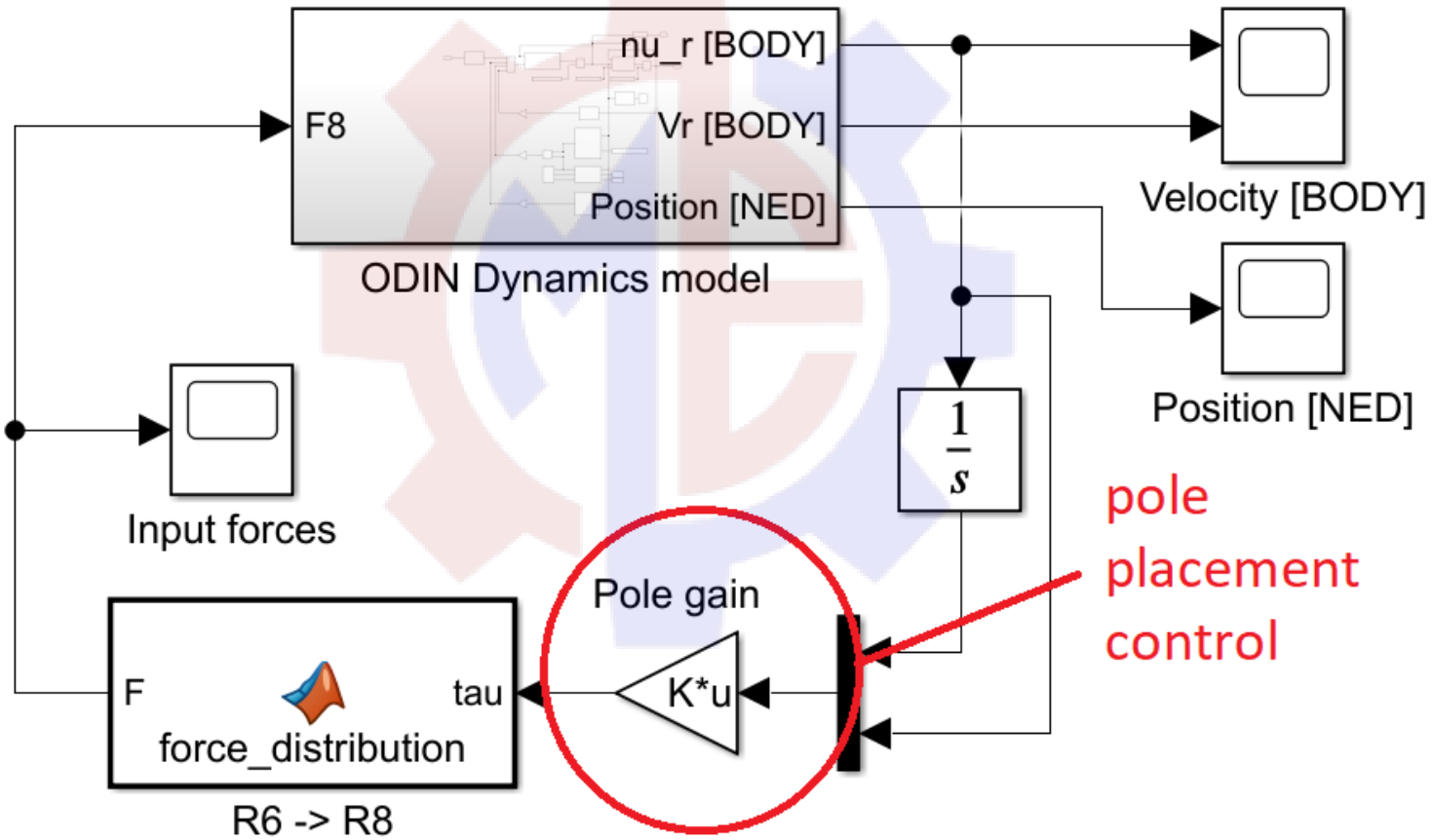
attack angle

design parameters

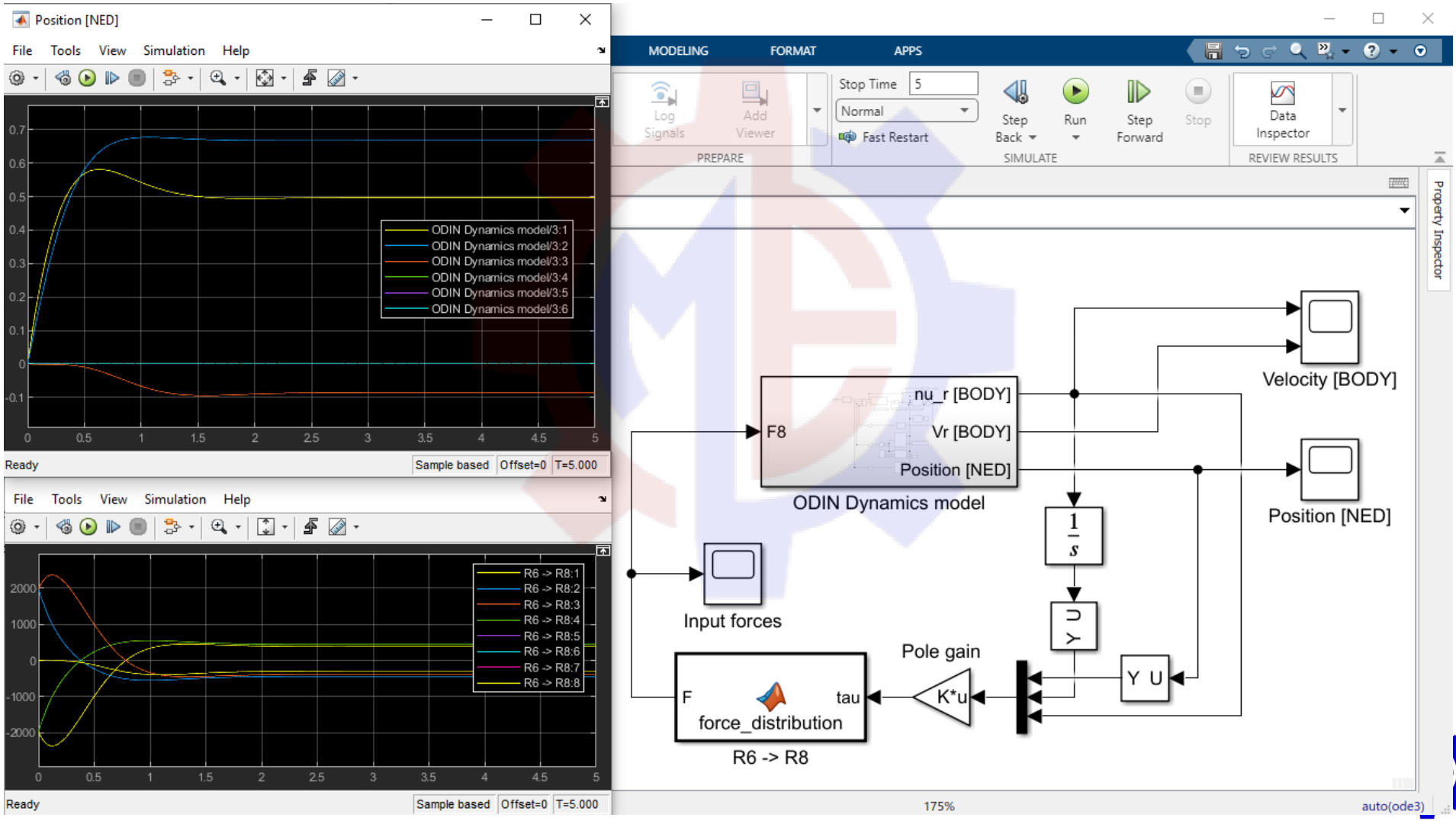
Added mass

```
function FA = added_mass(nu_r, Dnu_r)

ur = nu_r(4); vr = nu_r(5); wr = nu_r(6);

rho = 1025;
R = 0.5;

I3 = eye(3); % 3x3 identity matrix
O3 = zeros(3); % 3x3 zero matrix

% Added mass matrix MA
MA = rho * (2/3) * pi * R^3 * [I3, O3; O3, O3];

% Coriolis term of added mass CA(nu_r)
CA = rho * (2/3) * pi * R^3 * [
    0,   0,   0,   0,  wr, -vr;
    0,   0,   0, -wr,  0,   ur;
    0,   0,   0,  vr, -ur,  0;
    0,  -wr,  vr,  0,   0,   0;
    wr,  0,  -ur,  0,   0,   0;
   -vr,  ur,  0,   0,   0,   0];

FA = MA*Dnu_r + CA*nu_r;
end
```

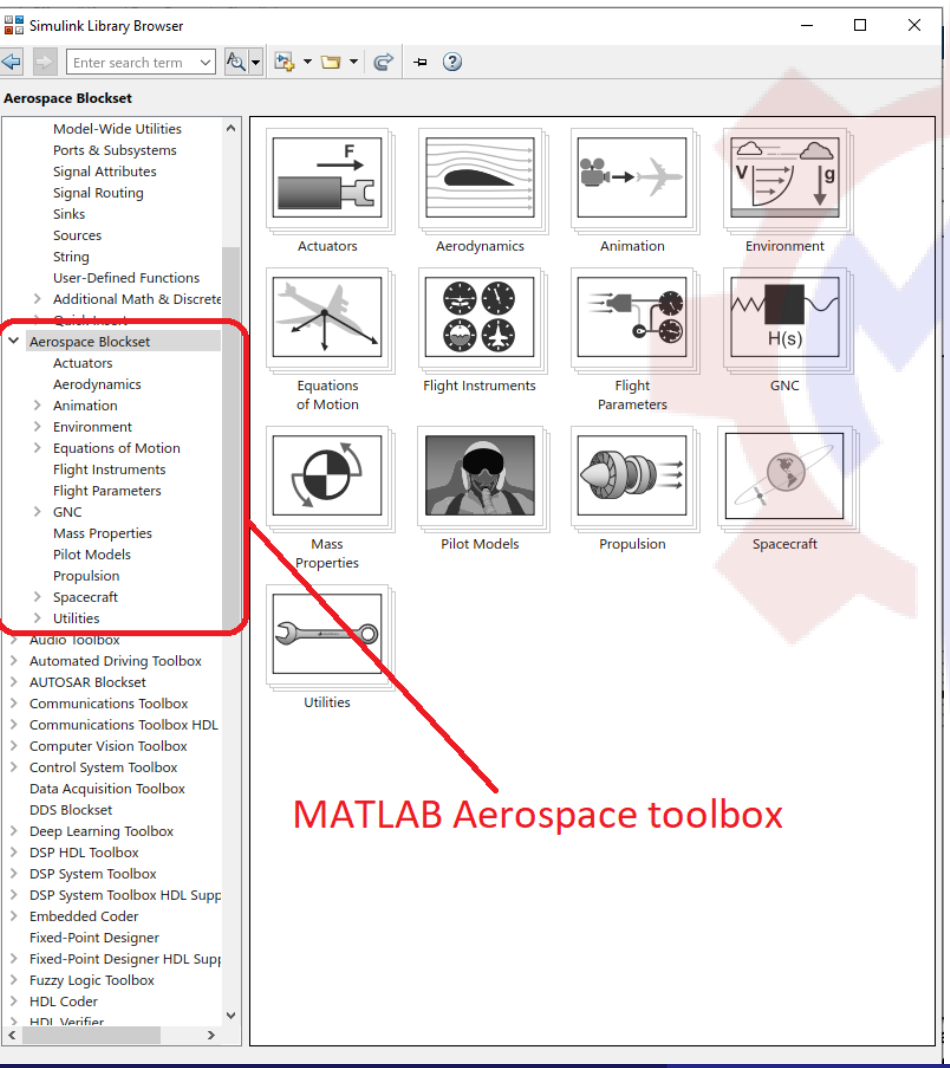
Simple control structure

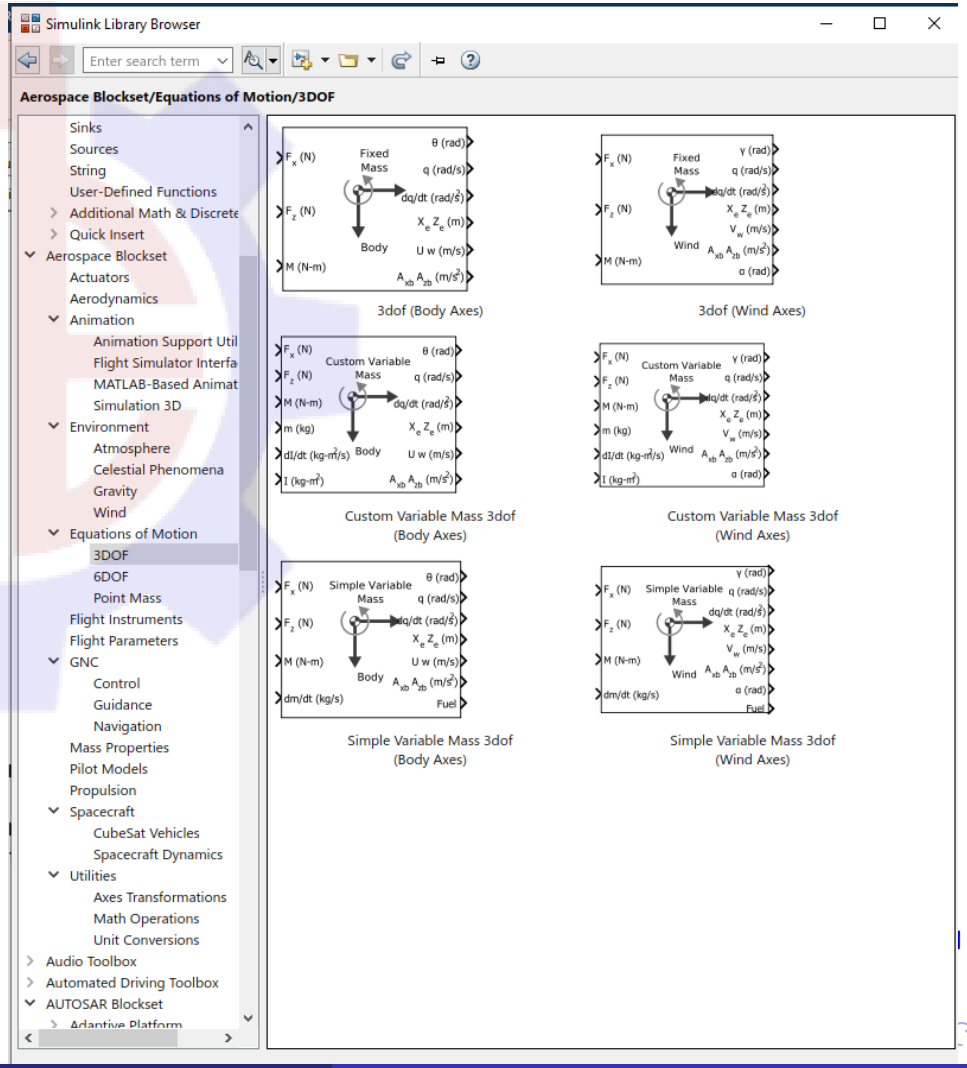


Simple control structure

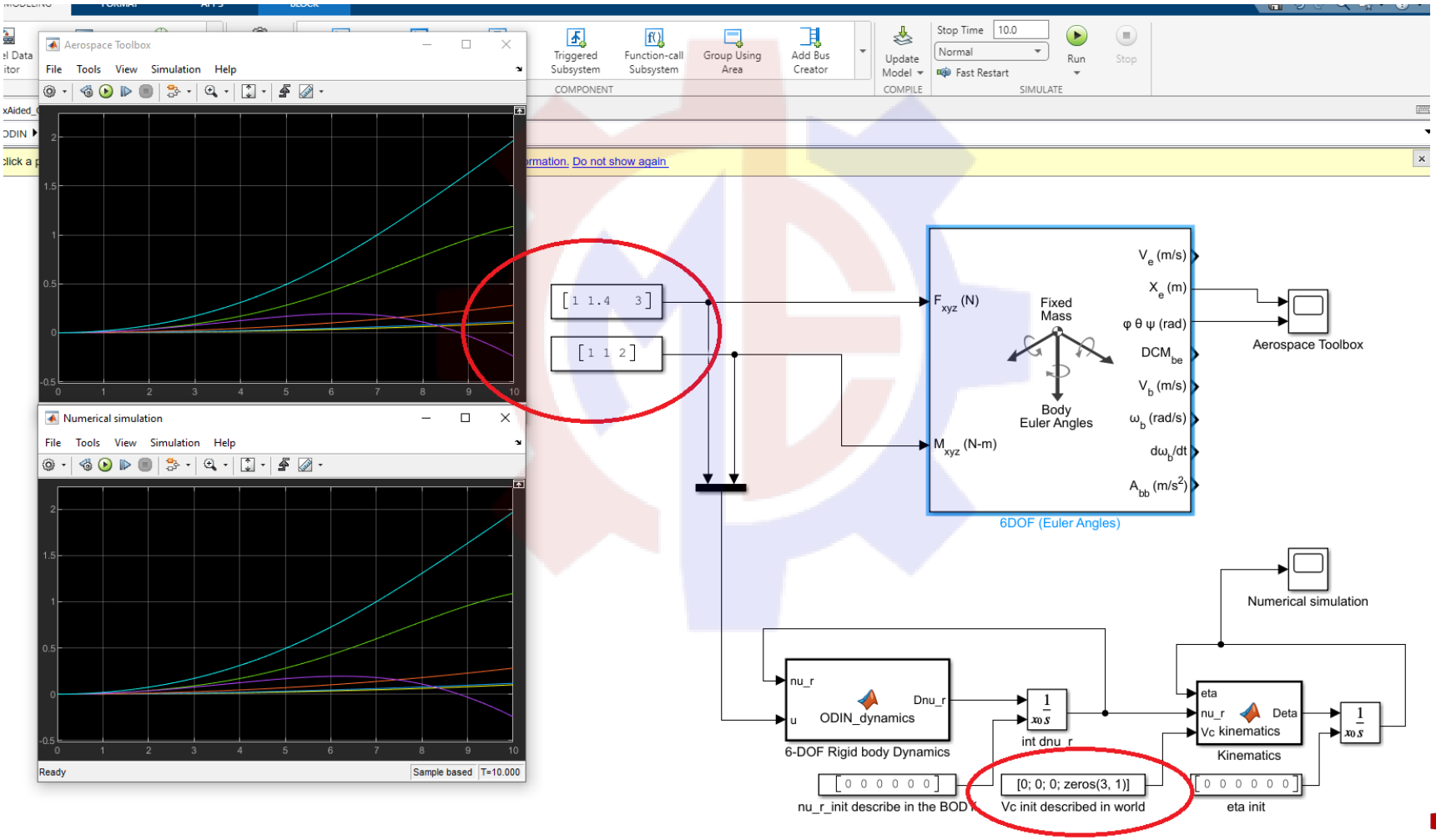


Aerospace Toolbox provides standards-based tools and functions for analyzing the motion, mission, and environment of aerospace vehicles.

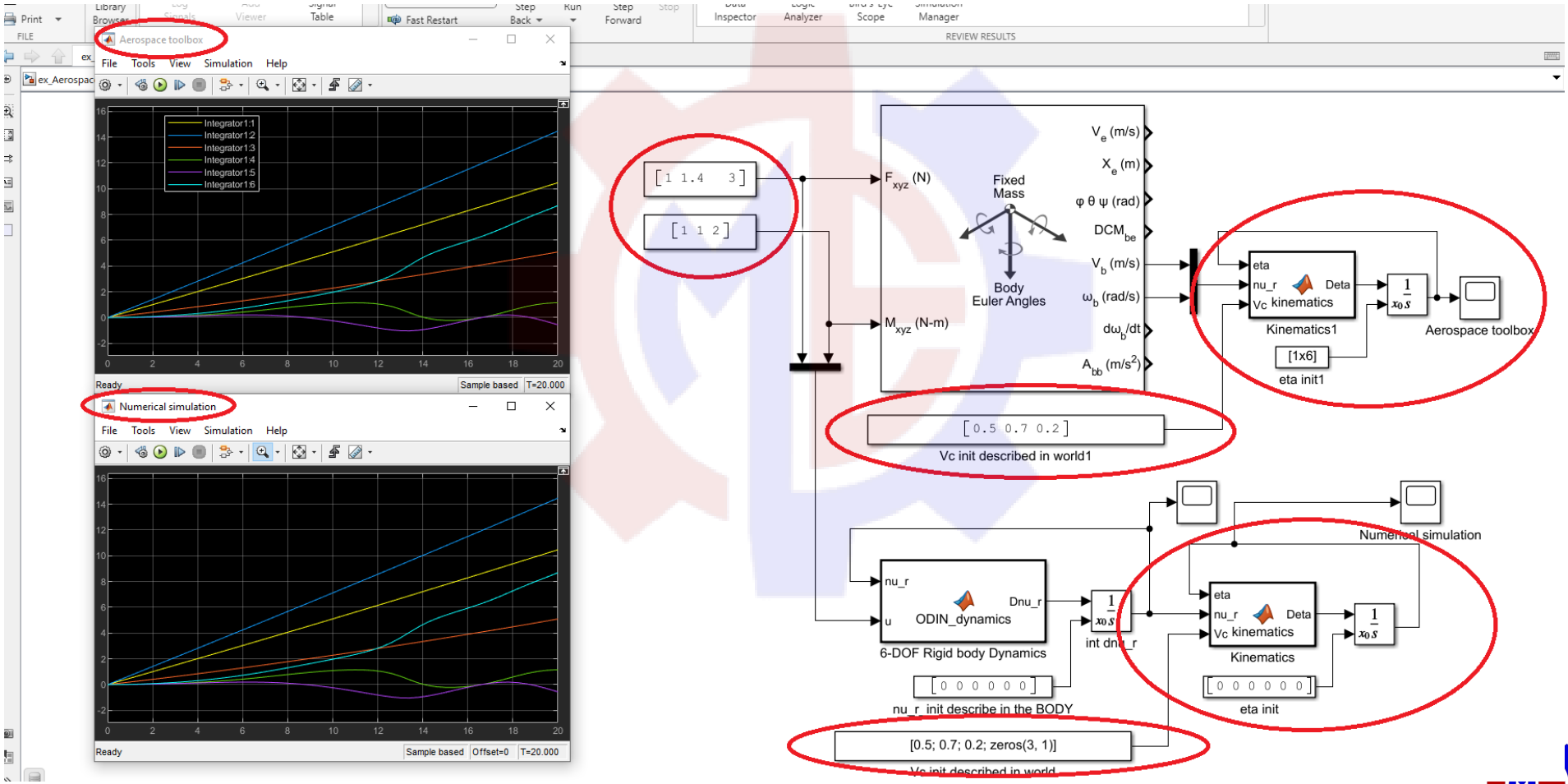




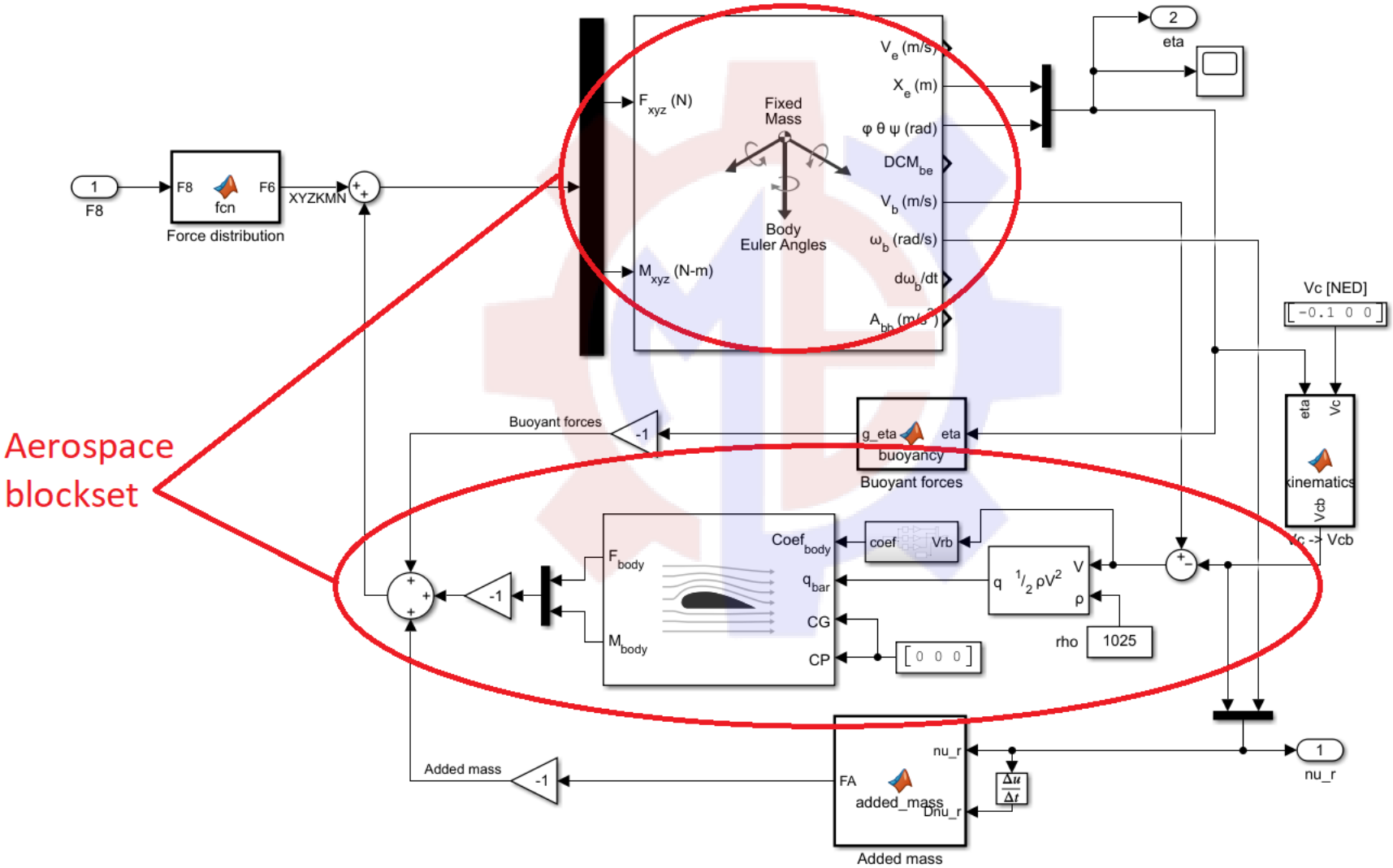
A comparison of Aerospace toolbox and traditional numerical simulation



Ocean current consideration.



Modeling AUV with buoyant forces, nonlinear damping, and added mass.



Simulation control structure

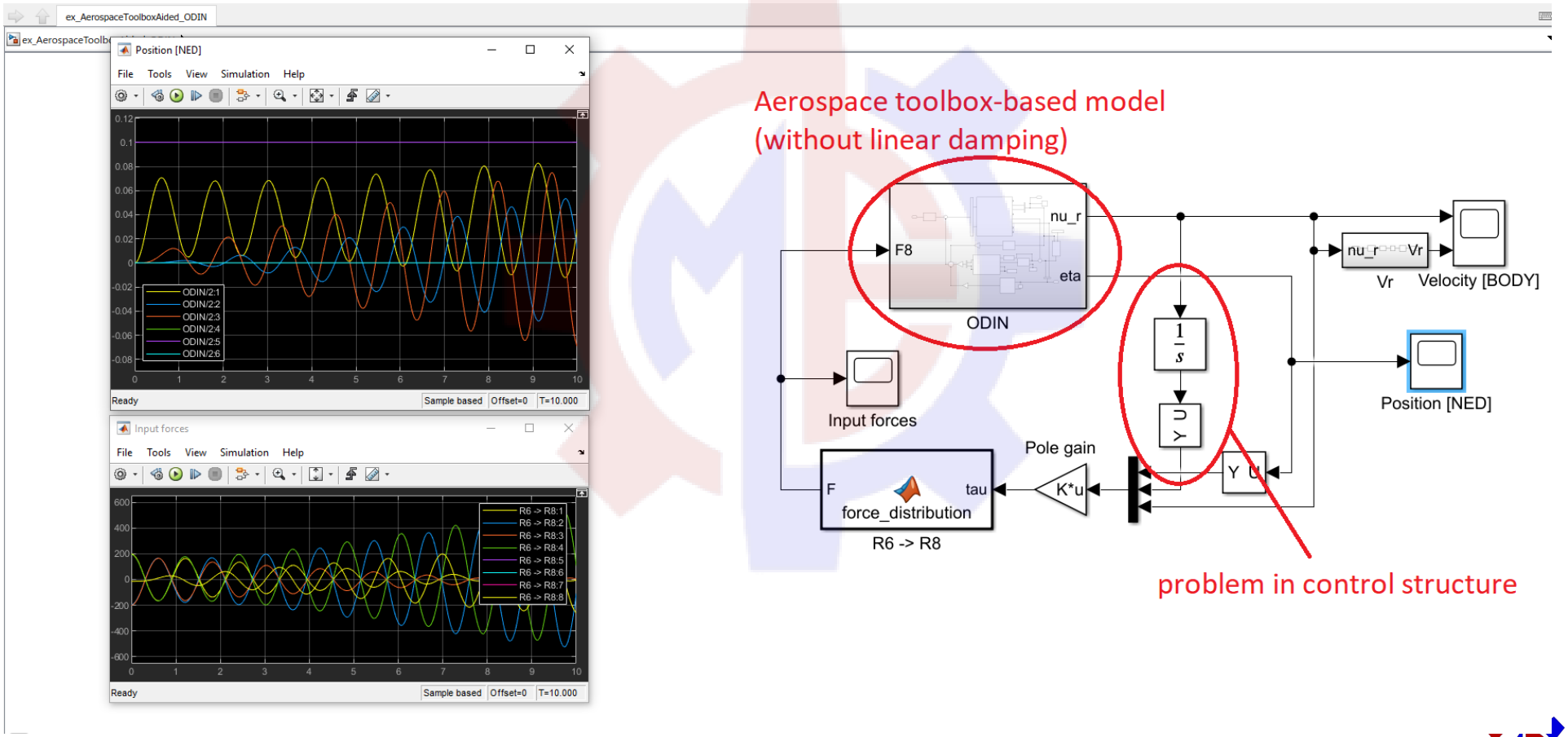
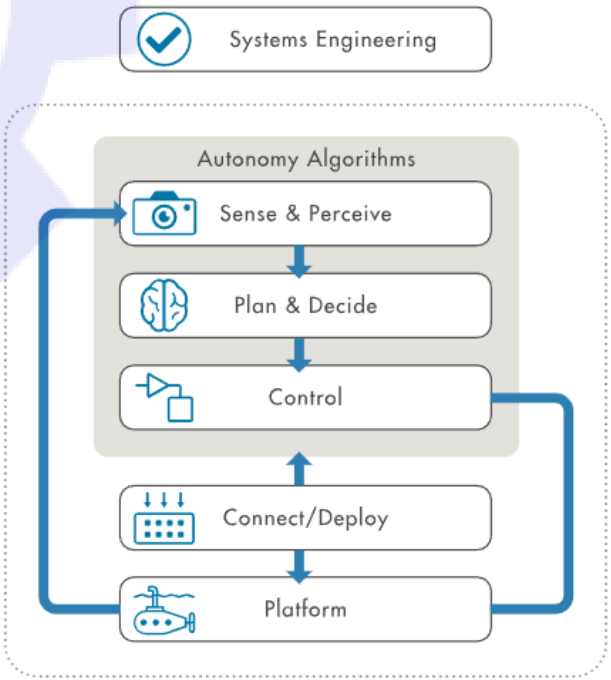


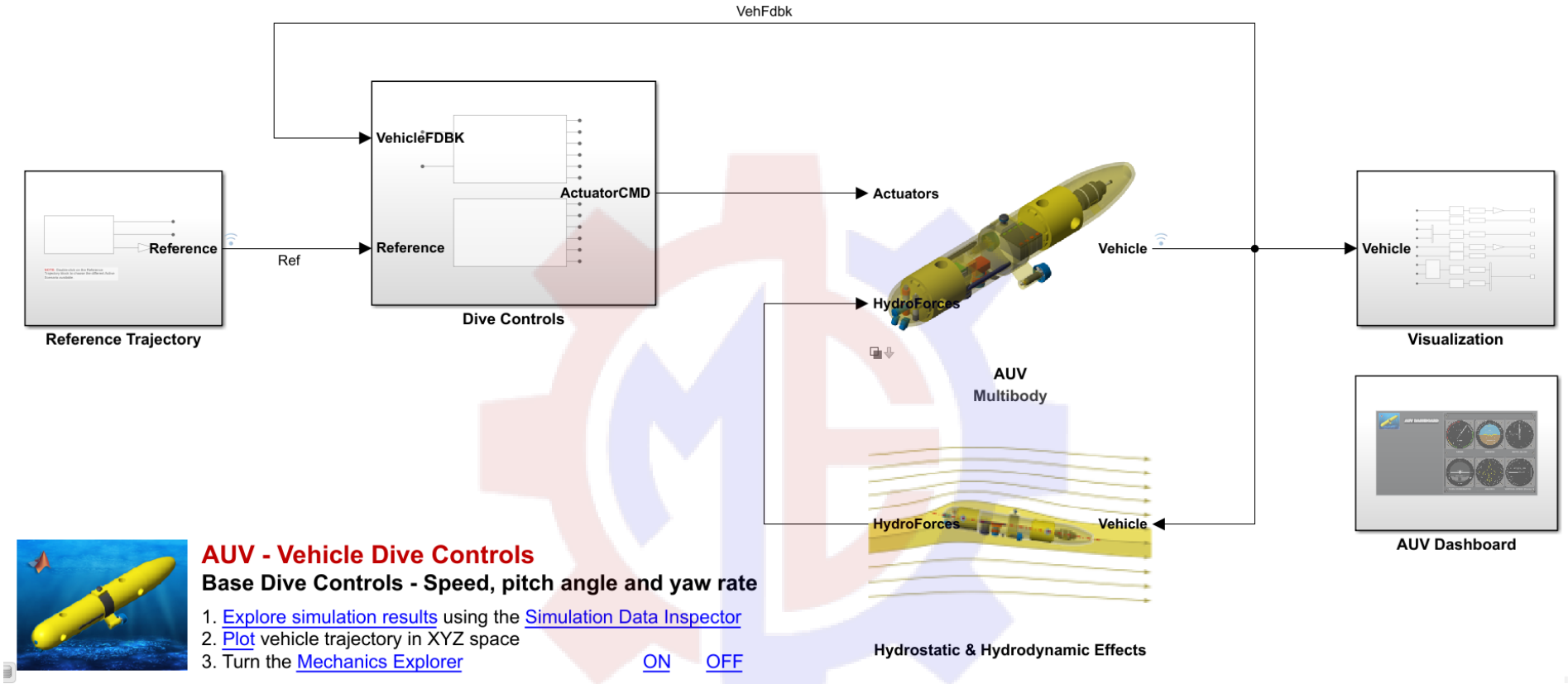
Table of Contents

Interdisciplinary teams can use MATLAB and Simulink as a common integration environment throughout the entire autonomous underwater vehicle workflow. From systems engineering to platform modeling, environment simulation, and autonomy algorithm design, Model-Based Design helps you reduce risk and build confidence in system performance well in advance of the sea trial.



Quasi-Physical Simulation for UV motion control

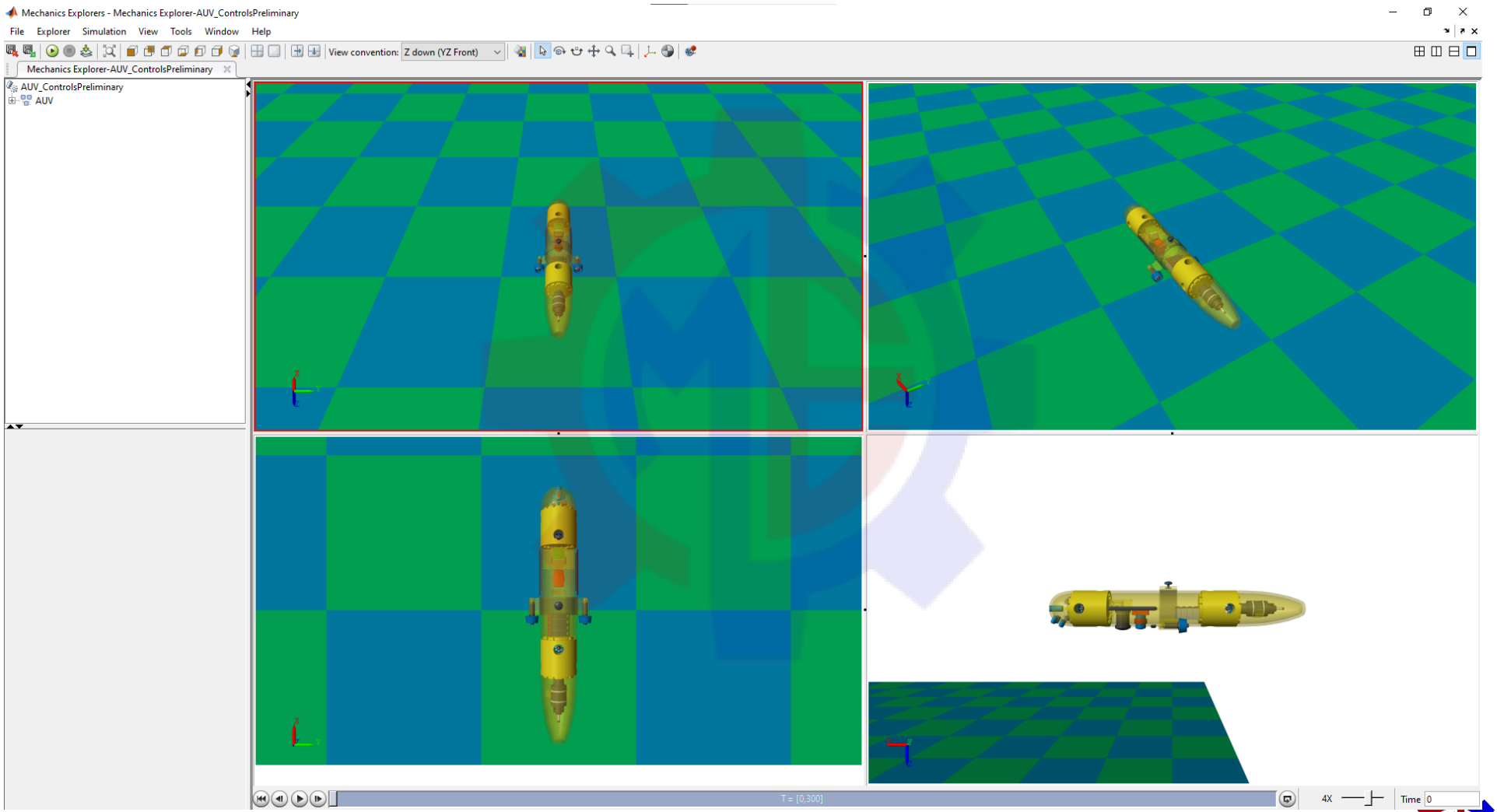
Mathworks AUV



Simspace Multibody is used to simulate the mathimatical model of the vehicle. Simspace Multibody ultilizes a combination of interconnected bodies (inertias), joints and constrants (relative DoFs) and force elements to solve the dynamics of the model

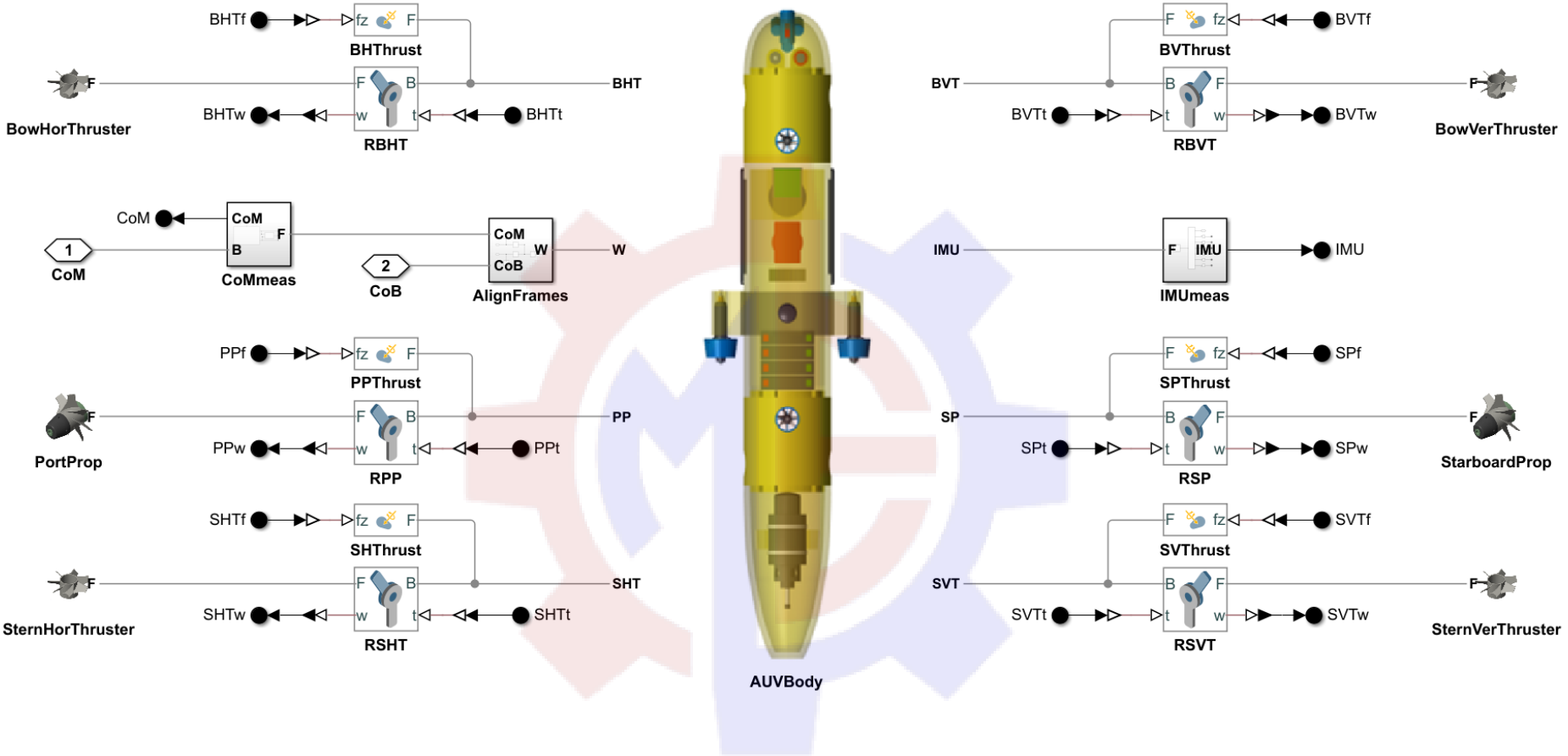
Quasi-Physical Simulation for UV motion control

Mathworks AUV



IMU located in the main hum, 6 actuators:

- BowVerThurster (vertical thurster located at the front of AUV)
- BowHorThurster (horizontal thurster located at the front of AUV)
- StarboardProp (Right Propeller)
- PortProp (Left Propeller)
- SternVerThurster (vertical thurster located at the back of AUV)
- SternHorThurster (horizontal thurster located at the back of AUV)



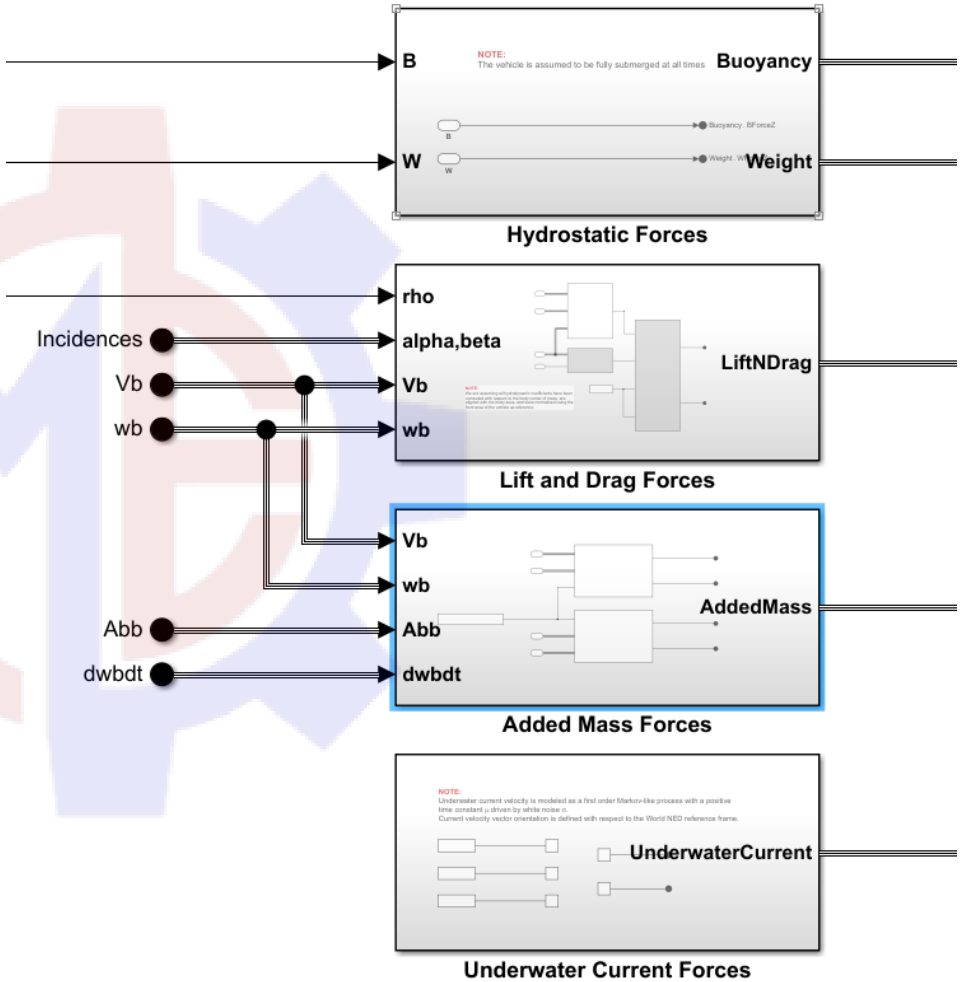
To describe the behavior of a multi-body system, we define the corresponding coordinates and attached forces

- Body frames can be created using body elements block with preset shapes and connected using rigid transform blocks

- Joint blocks connect the frames and impose kinetic constraints and determine how they move relative to each other

- The External Force and Torque represents forces or torques applied on the multibody model.

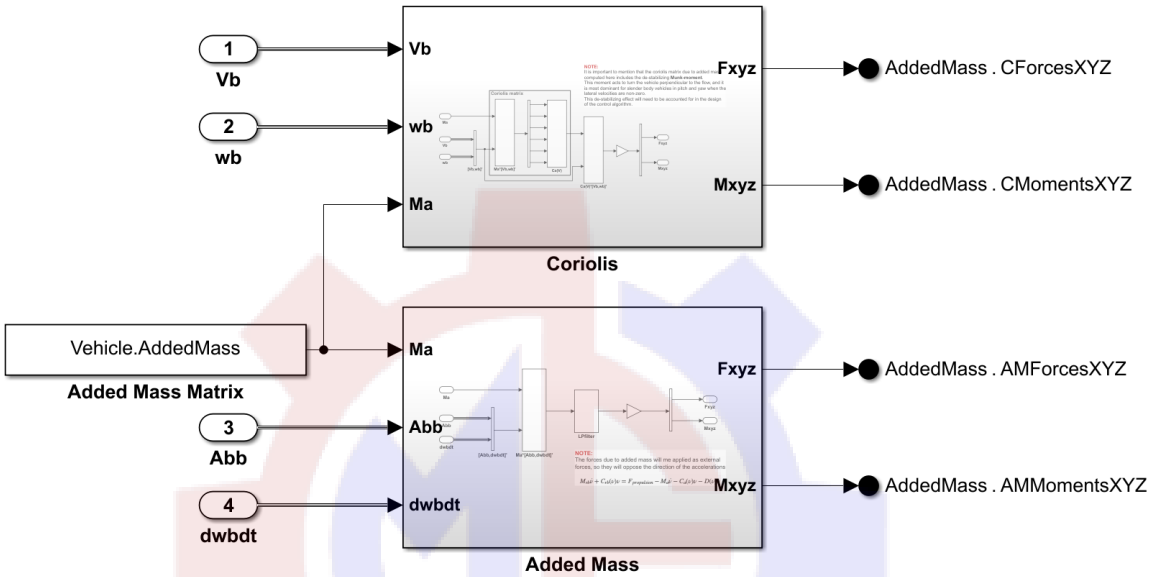
- File Solid Block assosiates individual part files from CAD formats to body blocks in the model.



Represents external forces of the marine environment.

Forces consist of:

- Gravity and buoyancy effects
- Lift and drag forces
- Added mass effects
- Underwater current forces



Mathematical model

$$(M_{RB} + M_A)\dot{\nu} + (C_{RB} + C_A)(\nu)\nu + D(\nu_r)\nu_r + \underbrace{g(\eta) + g_o}_{\text{hydrostatic forces}} = F_{propulsion} \quad (159)$$

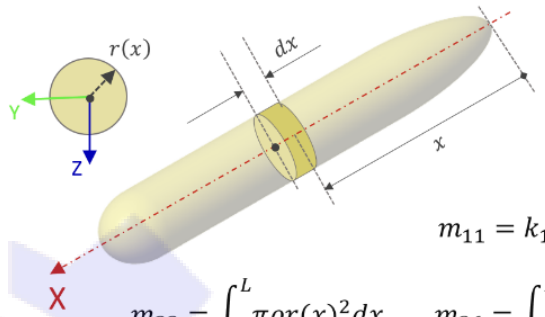
where M_A is the added mass matrix and C_A is the coriolis effect induced by the added mass

Added mass

The elongated myring shape of the hull results in

$$M_A = \begin{bmatrix} m_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 & 0 & m_{26} \\ 0 & 0 & m_{33} & 0 & m_{35} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{53} & 0 & m_{55} & 0 \\ 0 & m_{62} & 0 & 0 & 0 & m_{66} \end{bmatrix}$$

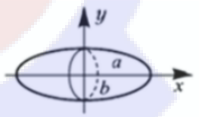
$$m_{22} = m_{53} \quad and \quad m_{55} = m_{66}$$



$$m_{11} = k_{11}\frac{4}{3}\pi\rho r_1 r_2^2$$

$$m_{22} = \int_0^L \pi\rho r(x)^2 dx \qquad m_{26} = \int_0^L x\pi\rho r(x)^2 dx$$

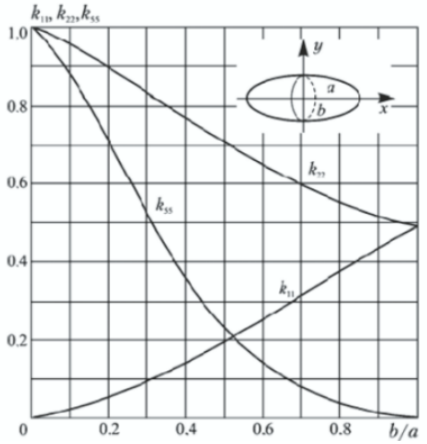
$$m_{66} = \int_0^L x^2\pi\rho r(x)^2 dx \qquad m_{35} = -\int_0^L x\pi\rho r(x)^2 dx$$



a = 1.0
b = 0.2

Non Dimentional value
$k_{11} = 0.05384$
$k_{22} = 0.85384$
$k_{55} = 0.89538$

$4/3\pi ab^2 = 0.167551$
$4/15\pi ab^2(a^2+b^2) = 0.034850$

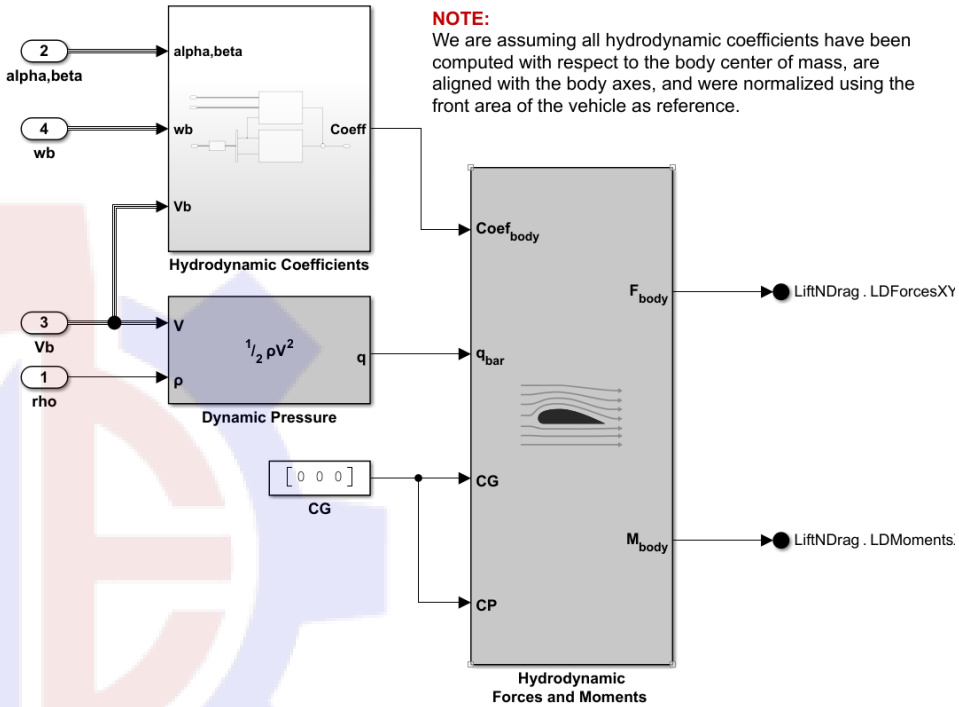


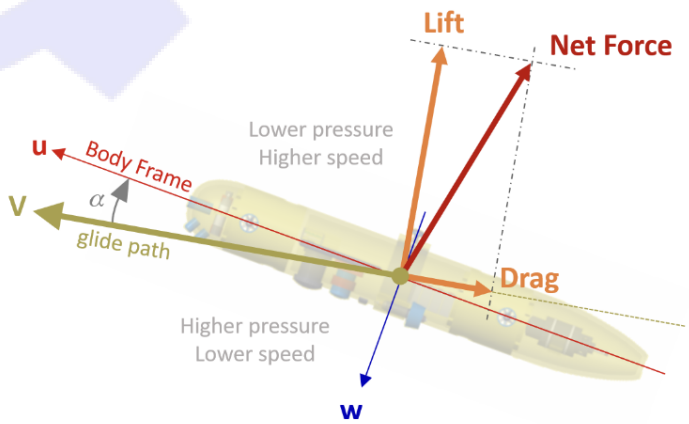
Drag and lift forces

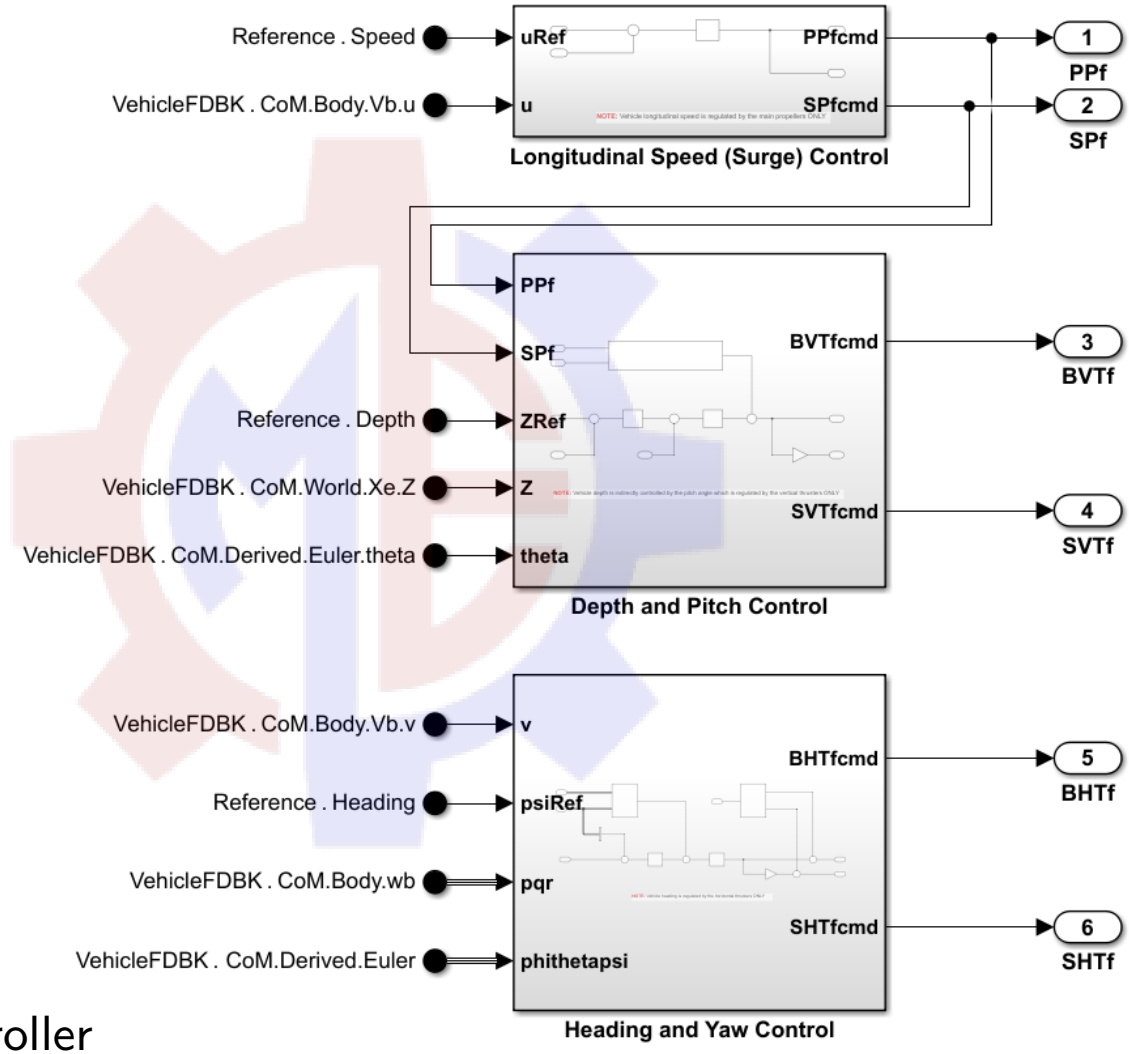
$$F_{drag} = \frac{1}{2}\rho V^2 C_d A$$

$$F_{lift} = \frac{1}{2}\rho V^2 C_l A$$

where C_d and C_l are normally from experiments and used in practice



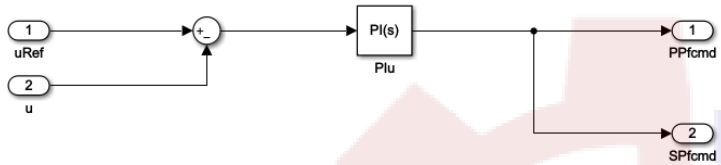




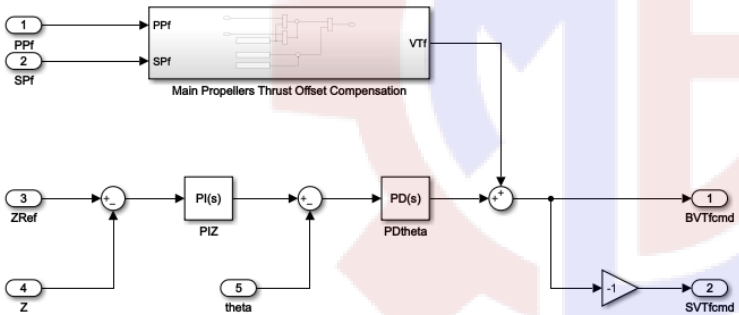
3 Controllers

- Surge Controller
- Heave and Pitch Controller
- Yaw Controller

- Surge Controller controls both propellers



- Heave and Pitch Controller controls vertical thursters



- Yaw Controller controls horizontal thursters

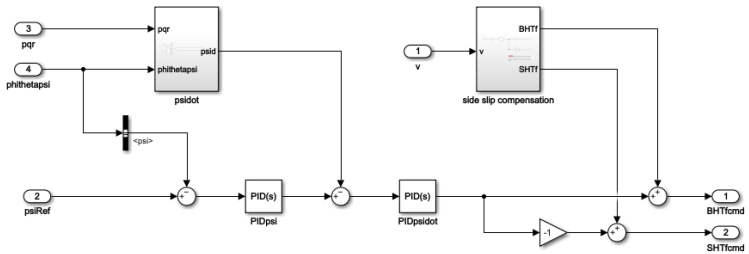


Table of Contents

An excample for GNC-integrated Quasi-Physical simulation

[Coming soon!!]

Thanks for your attention! Any questions?

Hope you slept comfortably!